SCSI Technical Reference Manual

Featuring SCSIdos 3.0

And AmigaDOS 1.3

Customer Service: (316) 267-0202
24 Hour BBS: (316) 267-1222
FAX: (G1/G2/G3) (316) 267-0111

Table Of Contents

Part I - SCSIdos 3.0 Instruction Manual

Introduction	Page 1
Acknowledgements and Credits	Page 2
Trademark and Copyright Notices	Page 2
C Ltd. SCSI Host/Controller Guaranty	Page 3
Hard Drive Guaranty	Page 3
General Precautions	Page 4
Quick Start Instructions	Page 5
C Ltd. SCSI Host/Controller Installation	Page 7
For Amiga A-500	Page 8
For Amiga A-1000	Page 9
For Amiga A-2000	Page 10
C Ltd. Hard Drive Installation and Testing	Page 11
C Ltd. Startup-Sequence	Page 12
Setting-Up An External C Ltd. Hard Drive	Page 13
Setting-Up An Internal C Ltd. Hard Drive	Page 14
Generic (non-C Ltd.) Hard Drive Installation and Testing	Page 17
Setting-Up An External Generic Hard Drive	Page 18
Setting-Up An Internal Generic Hard Drive	Page 20
Generic (non-C Ltd.) Hard Drive Configuration and Formatting	Page 23
Updating Existing Systems	Page 25
Using Your Hard Drive	Page 28
Installing Software on a Hard Drive	Page 29
The AmigaDOS Mount Command	Page 30
The SCSIdos DevInstall Command	Page 31
The Startup-Sequence	Page 33
MLed - The MountList Editor	Page 35
Formatting a Hard Drive	Page 41
When to Use the Low-Level Format Program	Page 42
Using the Low-Level Format Program	Page 44
FEdX (the Low-Level Format Program)	Page 50
Using the AmigaDOS Format Program	Page 62

Table Of Contents

Part II - SCSIdos 3.0 Technical Reference Manual

Introduction	TRM - 1
Glossary	TRM - 2
An OverView of SCSI Mass Storage Systems	TRM - 4
SCSI System Operations	TRM - 6
An Overview of SCSIdos Revision 3.0	TRM - 9
SCSIdos Programs and Files	TRM - 11
SCSIdos Sample MountList	TRM - 18
SCSIdos Sample DevSetup File	TRM - 19
Setting the SCSI Address	TRM - 33
Connecting Additional SCSI Devices	TRM - 34
Extending the SCSI/1000 Host/Controller	TRM - 37
Backing up Your Hard Drive	TRM - 38
Partitioning Your Hard Drive	TRM - 39
The Interleave Factor	TRM - 44
SCSInet	TRM - 47
C Ltd. SCSI Host/Controller Specifications	TRM - 49
Using the SCSI Host/Controller with Other SCSI devices	TRM - 51
Potential Problems and/or Unexpected Events	TRM - 52
Calling Customer Service	TRM - 54
Other Utility Programs	TRM - 55
The SCSI-Link Library	TRM - 58
INDEX	**TRM - 69**

Introduction

The Technical Reference Manual is not intended to be an instruction manual, rather it is a collection of information about Small Computer Systems Interfacing (SCSI) and the Amiga Computer. This manual will assume that the reader has a basic understanding of the Amiga computer and its operating software. The information provided in this manual is designed to assist developers, dealers and users to better understand the operation of the Amiga in relation to SCSI devices in general and Hard Disk Drive Systems in particular.

Due to the wide range of topics covered within this manual, little effort was put into maintaining a smooth flow of ideas from section to section. Neither was this manual designed as *'easy reading'*, in fact, it may take several readings of a section before the reader will understand the information contained therein. While some of the topics covered may seem trivial, they were included as they are the foundation for other weightier information and understanding of the more complex concepts is based on an understanding of their foundations.

While we are not in the business of education, if you have a real need to make use of the information contained in this manual need any assistance using or understanding the required information, you are encouraged to contact our Customer Service department. Our goal is to provide you, our Customer, with products that fill your needs. In most cases, our products will meet those needs *'out of the box'*, but sometimes Customers have needs that we didn't foresee and in most of those cases we can, with a bit of fancy footwork, still fulfill those needs as well.

GLOSSARY

A short course in SCSI terminology as used in this documentation.

SCSI - stands for Small Computer Systems Interface and is generally pronounced SCUZZY or called by the letters S C S I. SCSI is a standardized hardware bus design for connecting small computers and peripheral devices together so they can talk to each other. There is also a set of standardized SCSI software commands that provide a somewhat universal language that all SCSI devices can use to communicate with each other. Additionally, many SCSI devices may use their own (non-standard or extended) commands to control device specific functions such as formatting. Therefore even though a device is branded as a SCSI device, and will have the proper connectors to connect to other SCSI devices, the software that makes one unit operate may not access all of the functions available in other devices.

SCSI bus or SCSI network - refers to the standard hardware wiring scheme that is used to connect SCSI devices. The SCSI wiring standard includes a 50 conductor cable with standard 50 pin dual-inline box type connectors that connect devices in a daisy chain manner, and requires termination resistors at both ends of the chain.

Apple/Mac Pseudo-SCSI - refers to connecting scheme used with Apple's Macintosh SCSI devices. Unlike the true SCSI standard cables, which provide connections for all 50 of the connections specified in the SCSI standard and uses a flat 50 conductor flat ribbon type cable, the Pseudo-SCSI system uses a DB-25 25 pin connector at the SCSI Host unit and a 50 pin D-type connector at the SCSI Device Controller. This system generally uses a round, shielded 25 conductor cable which radiates less R.F. noise and is therefore more suitable in the home environment than the unshielded flat ribbon cable.

SCSI device - refers to any device that can be connected to the SCSI bus. SCSI devices come in three types, SCSI Host Adapters, SCSI Device Controllers and SCSI Host/Controllers.

SCSI Host Adapter - refers to a hardware device that connects to a computer system and provides the standardized connections that allow the computer to be connected to the SCSI bus as a Host unit. Some computers (not the Amiga) have SCSI Host adapters built into them.

SCSI Host - refers to a SCSI device that is a master controller over the devices on the SCSI bus. It is possible to have several SCSI Hosts connected to one SCSI bus, as the SCSI hardware and software standards provide for the necessary bus arbitration and collision protection to allow multi-tasking.

SCSI Device Controller - refers to a hardware device controller that is connected to the SCSI bus and controls one (or several) specific hardware devices. The SCSI bus requires that every SCSI device be assigned a unique unit number between zero and seven, therefore allowing a maximum of eight SCSI devices to be connected to the SCSI bus. Some hardware devices have SCSI controllers built in, in other cases an external SCSI device controller can be designed to control a

certain class of hardware devices. For example most Hard Disk Drives use what is called the ST506/412 standard for communications (these are your typical generic I.B.M. type Hard Drives) and others are available with the SCSI controller built-in. In the latter case you could connect the Hard Drive directly to the SCSI bus. In the first situation, however, you would need a SCSI device controller that was designed to connect to the SCSI bus and act as a controller for an ST-506/412 type device. SCSI device controllers generally allow control over several devices of one type, and some will provide for control of several device types. Devices currently available (**from C Ltd.**) with built in SCSI device controllers include Hard Disk Drives in sizes from 20 to 760 Megabytes, Tape Back-up units in sizes of 20 to 100 Megabytes, WORM (Write Once, Read Many) Optical Disk Drives, CD ROM disk units (not currently offered due to lack of available Amiga software), Optical Scanners and Laser Printers. Stand-alone SCSI device controllers are available to control one to eight ST-506/412 Hard Drives, one to four FLoppy Disk Drives, one to four Tape Back-up units and various combinations of the above.

SCSI Host/Controller - refers to a device that can function either as a SCSI Host Adapter or as a SCSI device controller.
This type of SCSI device is generally used with computers that are connected to the SCSI bus at the same time as other computers (multiple SCSI Hosts) and need to be addressed as SCSI devices so as to allow computer to computer communications. A SCSI Host/Controller is in reality two unique SCSI Devices (a SCSI Host and a SCSI Device Controller) that share the same connection to the SCSI Bus.

SCSI Initiator - refers to the SCSI device that initiates an action. Both SCSI Hosts and SCSI Device Controllers can act as Initiators and initiate SCSI commands. Since all responses to a SCSI command are directed to, and only to, the SCSI Initiator that originally issued the command, SCSI Initiators need not have an assigned SCSI Unit Number. This removes any limit on the number of SCSI devices that may reside on the SCSI bus and act as SCSI Initiators. Therefore it is possible to have a full load of seven SCSI Device Controllers on the SCSI Bus and still connect several computers (acting as SCSI Hosts) to the bus.

SCSI Target - refers to the SCSI device that is the target of a SCSI Initiator's command. SCSI Targets are addressed by their SCSI Unit Number, therefore only SCSI Device Controllers can be SCSI Targets since SCSI Hosts have no SCSI Unit Numbers assigned to them.

SUMMARY:

In summary, a SCSI system consists of one or more SCSI Hosts (computers) connected, by means of a standardized wiring scheme, to from one to seven SCSI Device Controllers which, in turn, may each control one or more devices such as Hard Disk Drives, Tape Drives, CD ROMs, Image Scanners, Printers, etc.

An Overview of SCSI Mass Storage Systems

Your C Ltd. SCSI Host/Controller will allow you to connect any type of SCSI mass storage device to your Amiga computer. There are currently several types of SCSI mass storage devices available from C Ltd. (and many other sources) including Hard Disk Drives, Floppy Tape Drives, Write Once/Read Many (WORM) Optical Disk Drives, Read Only (CD-ROM) Optical Disk Drives, several varieties of Tape Drives as well as some (cartridge and disk) Removable Media Hard Drives. C Ltd. even offers an SCSI Hi-Resoultion Page Scanner with Optical Character Regognition and a very fast 300 DPI SCSI Laser Printer. Most people will be primarily interested in the Hard Disk Drive systems, but any of the above devices can be added to your Amiga as required.

The C Ltd. SCSI Host/Controller has been designed with growth and expansion in mind. The hardware and software provided with your C Ltd. SCSI Host/Controller will allow you to connect, set-up and access information from any SCSI device. Though a Hard Drive and a Tape Drive may seem to be very different devices, the software provided with your C Ltd. SCSI Host/Controller allows AmigaDOS to deal with either device at a system level using the same commands that you are now using with the standard Amiga 3.5" floppy disks.

The reason that dealing with such diverse devices is such a simple matter is the consistent way in which C Ltd.'s SCSIDOS software treats these devices. When talking to SCSIDOS, AmigaDOS "sees" every device as nothing but a sequence of contiguous blocks of data. SCSIDOS relieves AmigaDOS of all housekeeping functions like keeping track of what track/cylinder etc. contains a specific block of data, what bad blocks are mapped out, where replacements for the mapped-out blocks are stored and how the blocks are scrambled by the interleave process. In simple terms, all AmigaDOS does is maintain its normal directory functions, and when it stores or retrieves information, AmigaDOS supplies SCSIDOS with a number that refers to the data block at which AmigaDOS thinks the data should be, asks SCSIDOS for as many blocks of data it needs and leaves the task of actually finding where on the disk or tape the data is really stored up to SCSIDOS.

When operating under SCSIDOS, the HighCyl, LowCyl, Surfaces and BlocksPerTrack data provided in the MountList (see MountList File) is rendered meaningless except that the combination of the values MUST equal the TOTAL number of data blocks that the device is to contain. The AmigaDOS format program (see AmigaDOS Format) is used to put the formatting information that AmigaDOS expects to see on to the device. This information tells AmigaDOS the size (in total number of blocks) of the device and the starting block, usually zero, except in the case of a partitioned (see Partitioning your Hard Drive) device, of the device. The only value that AmigaDOS cares about is the TOTAL number of data blocks. That value must match the value that AmigaDOS calculates using the HighCly, LowCyl, Surfaces and BlocksPerTrack values from the Mountlist file.

An example of this might be using a Tape Drive to Back-up a Hard Drive. Let's propose a system with a C Ltd. 24 Meg. Hard Disk and a 30 Meg Tape Drive. The Hard Drive has 4

surfaces, a LowCyl of 0 and a HighCyl of 647 for a total of 648 Cylinders and a BlocksPerTrack value of 18. The Hard Drive then has (4 * 648 * 18 = 46656) 46,656 Data Blocks, with 512 bytes per block, yielding a total storage capacity of 23,887,892 bytes or 23.89 Megabytes. The Tape Drive, however, has only 1 surface with a LowCyl of 0 and a HighCyl of 11 for a total of 12 cylinders and 4800 BlocksPerTrack. The Tape Drive then has (1 * 12 * 4800 = 57600) 57,600 Data Blocks, with 512 bytes per block, yielding a total storage capacity of 29,491,200 bytes or 29.49 Megabytes. Using the AmigaDOS format program, we would format the Hard Drive with the values as shown above, but we would also format the Tape Drive with a MountList that reflects the same values, even though they don't reflect the actual values relative to the Tape Drive unit the total number of data blocks specified is less that the total number of data blocks available on the Tape Drive so this is a safe practice. This would make both devices contain the same number of TOTAL Data Blocks. (4 * 648 * 18 = 46656 = 1 * 12 * 3888) Then all we need to do is maintain identical mountlists for the two devices, we can use any desired values for Surfaces, and BlocksPerTrack as long as they are the same, and we can use any values for LowCyl and HighCyl as long as the total number of Cylinders is the same. (The AmigaDOS DiskCopy command DOES compare Surfaces, BlocksPerTrack and total Cylinders and these values must match for the command to work.) Now with the proper MountList information in place, we can mount the devices and use the standard AmigaDOS DiskCopy command to do a quick, complete back-up or restore of the Hard Drive to/from the Tape Drive.

The C Ltd. CAP command (see Other Utilities) can provide an easy method of determining the formatted capacity (in sectors/blocks) of any formatted SCSI device. The CAP command requires that the SCSI device must be properly Low-Level formatted, but does not require the unit to be AmigaDOS formatted.

The consistency of this system allows SCSIDOS the universality necessary to interface virtually any SCSI device to the Amiga and allows the user the comfort of being able to use the familiar standard AmigaDOS commands to access the device. There are, however cases where the user may want to access special features and/or functions of some SCSI devices that are not accessible through the standard AmigaDOS interface. (For example, a Tape Back-up Utility that will back-up a Hard Drive that is larger than the capacity of one tape.) For these cases, we have provided the SCSI-Link Library (see Other Utilities) that should allow anybody who has a reasonable amount of programming experience to communicate directly with devices on the SCSI bus/network.

SCSI System Operations

The small box that plugs onto the side your Amiga (or the circuit board in the case of an Amiga A-2000) is actually a full SCSI Host/Controller that can talk to up to seven SCSI Device Controllers, and can (and in the near future will be able to) be addressed by other SCSI Hosts as a SCSI Device Controller. If you have an external Hard Drive, the external Hard Drive box (if you own an AM-22, AM-24, AM-33, AM-44, AM-50 or AM-80) contains two pieces of hardware, an Adaptec SCSI to ST-506/412 Device Controller that can control up to two ST-506/412 Hard Disk Drives, and one of the standard ST-506/412 Type Hard Drive mechanisms in the appropriate capacity. The two components are shipped from the factory with the proper connections for both true SCSI and Apple/Mac Pseudo-SCSI wiring configurations, but only the 25 conductor Apple/Mac Pseudo-SCSI cable is provided. If you have an internal Hard Drive, it could be either an SCSI Hard Drive with the SCSI device controller imbedded (built-in) in the Hard Drive's electronics package, or an ST-506 Hard Drive, in which case the SCSI to ST-506 device controller will be attached to the C Ltd. SCSI Host/Controller.

Using the Amiga system you now own as an example, here is how a typical SCSI system would operate. You as the computer operator request some information from the Hard Drive that is connected to your Amiga on the SCSI system. The CLtd.device driver software (CLtd.device is the file name of the driver software that has been provided with your C Ltd. Host/Controller), working together with the standard AmigaDOS file system in your Amiga converts your request into a series of commands that conform to the standardized commands that are used by the SCSI system. These commands are then sent to the C Ltd. SCSI HOST/CONTROLLER. Processing these commands in its HOST mode of operation, the C Ltd. SCSI HOST/CONTROLLER initiates the proper command sequences to cause the SCSI CONTROLLER that is connected to your Hard Drive to activate the Hard Drive, retrieve the requested information and pass it back to the C Ltd. HOST/CONTROLLER which then passes the information back to your Amiga. So, from the perspective of the SCSI system, the HOST is an active device, it initiates commands and supervises activities on the SCSI system. Whereas, the SCSI Hard Disk Drive CONTROLLER is a passive device, it only acts on commands sent to it over the SCSI system and does nothing without being told to.

SCSI CONTROLLERs, like the CONTROLLER on your Hard Drive, can come in many forms. Most CONTROLLERs are designed to perform a specific task, as in this case it controls a Hard Disk Drive. A CONTROLLER is like a translator, it listens to a standard set of commands from the SCSI system, then translates them into the specific commands used to control the device to which it is connected. This is a very simplified view of what a SCSI CONTROLLER does, but it will give you the general idea. As we said, SCSI CONTROLLERs can come in many forms, for instance a SCSI CONTROLLER can be a stand-alone circuit board (like the ADAPTEC ACB 4000 - referred to elsewhere in this manual) that can be used to connect virtually any ST-506 (I.B.M. standard) Hard Drive to a SCSI system, or they can be designed right into the normal electronics of a Hard Drive system (normally referred to as a 'built-in' or 'resident' SCSI CONTROLLER) so as to allow the Hard Drive to be directly connected to an SCSI system.

The one thing that all SCSI CONTROLLERs have in common is that each must have a unique UNIT number. (UNIT numbers are discussed in several other sections of the C Ltd. SCSI Technical Reference Manual.) On a SCSI system the HOST is always assigned as UNIT # 0, there are up to eight UNITs allowed on the SCSI system, so the other CONTROLLERs must have the ability to be assigned as UNITs 1 through 7.

The C Ltd. SCSI Card is a HOST/CONTROLLER. A HOST/CONTROLLER, simply put, has the ability to act as both a HOST and a CONTROLLER. A HOST/CONTROLLER uses the UNIT # 0 when acting as an SCSI HOST, but also can be assigned an additional UNIT # (1-7) that is used when it is acting as a CONTROLLER. That's nice, but what is the advantage to you? Well, as your system exists today, there is none. But, this system allows a lot of potential for future growth. The following example takes a look into ***your*** future:

A little future music, please....

Its early 1990, you discover that you just can't keep up with your booming desktop publishing/video business with just one computer. Your complete development and design system is on your one lonely Amiga, it needs help! You could change to a multi-user mini-computer, but all of your data and programs are in Amiga format and there are some features of the Amiga that you are very reluctant to give up, not to mention the six-figure cost factor. You call your Amiga dealer for help. Your dealer says, "Gee, Steve, how many artists' stations do you want to support?" You answer, "At least three right now, but it looks like I may need six by next year. Your Dealer says, "You've got a C Ltd. SCSI Hard Drive System, don't you?" "Yup." you say. "No problem," he says, "I'll be over this afternoon and bring two more computers with me!"

How was the Dealer able to solve Steve's problem so easily? Because Steve's C Ltd. SCSI Hard Drive System had a HOST that was able to be a CONTROLLER as well. Normally, having two computers as HOSTs on the same system would pose a major conflict, because they would both want to be UNIT # 7 and there would be a conflict whenever a non-HOST controller was trying to send data back to the originating HOST (which UNIT # 7 am I talking to?), but the built-in arbitration of the SCSI system and the full SCSI arbitration ability of the C Ltd. SCSI HOST/CONTROLLER were designed to overcome just such problem situations.

Additionally, how does one computer (HOST) talk to another? To overcome this problem, your C Ltd. SCSI HOST/CONTROLLER has provisions to set its device number to any UNIT # from 0 to 7. When it is inactive, it looks to the SCSI system just like any other DEVICE CONTROLLER. So, another Amiga with a similar SCSI HOST/CONTROLLER could issue commands and get data with no conflict. Only when the C Ltd. SCSI HOST/CONTROLLER is actually called upon to take some action does it shed its CONTROLLER disguise and become an active HOST. You could actually have up to seven Amiga Computers on-line using only one (large, we would hope) Hard Disk Drive. Each Amiga could then function as a complete stand-alone work station for jobs like Word Processing or Forecasting, but be tied directly in to the central database when it needs to access data from the mailing list or or get some fonts or clip-art out of the main data bank.

The C Ltd. SCSI HOST/CONTROLLER not only can look like an SCSI CONTROLLER, it can act like one! Because the C Ltd. SCSI HOST/CONTROLLER is a TRUE SCSI CONTROLLER, it will allow other HOSTs on the SCSI system to treat the Amiga Computer it is attached to like any other SCSI device, that means that one Amiga will actually be able to retrieve programs and/or data from the floppy drives connected to another Amiga, or leave electronic mail in the RAM disk area of another Amiga. (NOTE: A security system, built into the system software, would normally prevent deleting or writing to another persons files.)

NOTE: The networking software required to allow the computer to computer communications described above is now being developed and should be available in the near future, but *the existing software that you now have* (ie SCSIdos 3.0) *is all that you need allow several computers to retrieve data simultaneously from one or more hard drives, or to communicate with a central SCSI Laser Printer.*

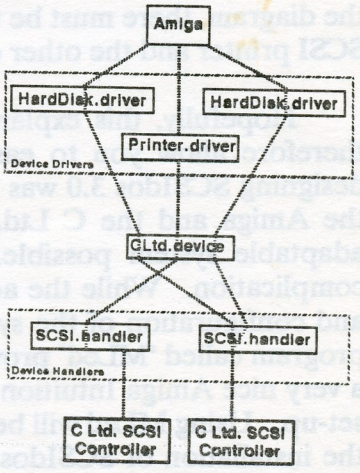
WARNING: Many SCSI HOSTs are **NOT** able to act as SCSI CONTROLLERs as well. If you see the possibility of future expansion in this direction, you made the right choice when you purchased the C Ltd. SCSI Host/Controller as the center of your Hard Drive System. But be sure that other future additions to your system can comply with your projected system expansion needs.

Additional Warning: It has come to our attention that some SCSI Adapters (including the Commodore 2090 and 2090a) do not allow the Computer's Host Address to be set at any position other than SCSI address 7, and these non-ANSI Standard SCSI adapters also do not conform with the published ANSI standard for SCSI bus arbitration. Because these non-standard devices do not have selectable SCSI Addresses and do not arbitrate properly they **SHOULD NOT** be used in any multi-host system. Additionally, it will be impossible for computers fitted with these non-standard devices to allow multiple Host computers to Auto-boot from a single, central Hard Drive.

Overview of SCSIdos™ Revision 3.0

SCSIdos 3.0 is the most recent (as of 10/88) version in the SCSIdos 3.0 series. SCSIdos 3.0 (and its revisions) was developed to replace C Ltd.'s previous SCSIdos 2.0 series and has been specifically enhanced to take advantage of the improvements in Commodore's AmigaDOS 1.3 operating system, specifically including the Fast File System (FFS) disk storage format. SCSIdos 3.0 is a modular software interface constructed to provide a bridge between two hardware systems, the Commodore Amiga computer and the Small Computer Systems Interface (SCSI) Bus. Additionally with its modular design, SCSIdos 3.0 can provide maximum flexibility, allowing easy customization of the software interface thereby facilitating its use with atypical SCSI interfaced devices (like page scanners, digitizers and Optical Disks) as well as Multiple Host Computer Systems.

To establish flexibility without sacrificing speed or code size, the interface software has been divided into separate programs also called tasks, running simultaneously, that perform the needed functions. One task, called the DRIVER, is the task that talks to the Amiga on one side and the next task on the other. The second task is called the DEVICE, it talks to the first task (the DRIVER) on one side and the third task on the other. The third task, the HANDLER, talks to the DEVICE on one side and to the SCSI bus on the other. The diagram on the right provides a graphic representation of how this system looks and works.



C Ltd. SCSIdos 3.0 Flow Chart

Referring to the diagram, notice that there are several device drivers, one printer driver and two hard disk drivers. There is a device driver (TASK) for EVERY device that is mounted on your system. The device drivers actually contain the settings of the various options and other information that applies to that particular device. The device drivers also act like filters that get data from the Amiga's operating system and filter out any un-needed or undesirable data. In the case of a hard disk driver, this might reflect the fact that the mounted DEVICE has been selected as write protected, so the device driver will reject any attempt to write to the device and will return a 'Device is Write Protected' error message to the Amiga operating system whenever it attempts to write data to the DEVICE. In the case of a printer driver this filtering action would intercept the generic Amiga printer escape sequences and translate them into the specific data that the printer requires to accomplish the action specified by the escape sequence. The device driver also specifies which DEVICE HANDLER should be used with its DEVICE. Again, due to the modular design of SCSIdos 3.0, device drivers can be written to deal with ANY type of device and then these device drivers can be easily added to the system without affecting the operation of other devices or requiring a total rewrite of SCSIdos just to add another type of non-standard device.

The next link in the chain is the CLtd.device. It is the clearing house where most of the work is done. The CLtd.device initiates the device drivers and the device handlers as tasks and keeps track of their activities. The CLtd.device also implements the options that apply to each of the different DEVICES (as contained in their respective device drivers) and arbitrates communications between the various tasks and handlers. There is only one CLtd.device active at any time.

Again referring to the diagram, notice that there are two device handlers and that one is connected though the CLtd.device to the Printer.driver while the other is connected through the CLtd.device to both of the HardDisk.drivers. The actual number of handlers is determined by the number of C Ltd. SCSI Host/Controllers that the computer has installed on/in it, there must be one SCSI.handler defined for each C Ltd. SCSI Host/Controller. Therefore, in the system in the diagram, there must be two C Ltd. SCSI Host/Controllers in the computer with one going to a SCSI printer and the other connected to one or more Hard Disks.

Hopefully, this explanation will help you to understand the way SCSIdos 3.0 works and therefore allow you to easily expand your system as the need arises. Our primary goal in designing SCSIdos 3.0 was to build an easily extensible, very flexible software interface between the Amiga and the C Ltd. SCSI Host/Controller that would provide its users with the most adaptable system possible. This tremendous flexibility comes at the price of some added complication. While the actual use of SCSIdos 3.0 is totally transparent to the user, the set-up and configuration of the system could have been be quite cumbersome if it had not been for a program called 'MLed' provided for us by Steve Tibbett (of VirusX and DiskX fame) which uses a very nice Amiga Intuition Interface, complete 'HELP' screens and greatly simplifies the system set-up. Using MLed will be covered next immediately following the next section which deals with the installation of SCSIdos 3.0, but for those of you who 'just have to know it all', a complete, commented listing of a sample SCSIdos 3.0 MountList and a sample DevSetup file is available at the end of this section.

If you are using a C Ltd. Hard Drive, you MUST use only the new SCSIdos 3.0 software provided on the Boot disk to run all partitions of the Hard Drive. You cannot run some of the Hard Drive under old versions of C Ltd.'s SCSIdos while you are using the new version on other parts as the two versions of the software will collide causing system failures.

NOTE: Reformating of your Hard Drive is not necessary to use SCSIdos 3.0, however, reformating may allow you to gain more speed by changing the interleave factor to take advantage of our higher speed operation. See the *'Low-Level Format'* section elsewhere in this manual.

SCSIdos™ Programs and Files

The files that are new and/or changed from all previous versions of SCSIdos are:

C:UpdateML
C:Checkfree
C:defdisk
C:devinstall

DEVS:CLtd.device
DEVS:HardDisk.driver
DEVS:SCSI.handler
DEVS:DevSetup
DEVS:MountList

S:Auto-Start
S:HardDrive.Startup-Sequence
S:Startup-Sequence
S:UserNotes
S:Floppy.Startup-Sequence
S:NewLook
S:typenote

Libs:SCSI-Link.library

MLed/Mled.help
MLed/MLed
MLed/MLed.info

Utilities/FmtData/#?

Utilities/HardDrives/#?

Utilities/1.a
Utilities/2.a
Utilities/3.a
Utilities/4.a
Utilities/5.a
Utilities/6.a
Utilities/Check
Utilities/Clear
Utilities/clearscsi

Utilities/CLtd-Format 1
Utilities/CLtd-Format 3
Utilities/Errors.info
Utilities/FEdX.info
Utilities/formatter.info
Utilities/Inquir
Utilities/Inquire.info
Utilities/scsicheck
Utilities/Size
Utilities/1
Utilities/2
Utilities/3
Utilities/4
Utilities/5
Utilities/6
Utilities/cap
Utilities/Check.info
Utilities/Clear.info
Utilities/CLtd-Format 0
Utilities/CLtd-Format 2
Utilities/Errors
Utilities/FEdX
Utilities/Formatter
Utilities/GenFormat
Utilities/Inquire
Utilities/readerr
Utilities/SCSIcom
Utilities/Size.info

Also included on the SCSIdos update utilities disk ONLY:

Update_To_3.0
Update_To_3.0.info

File by File Documentation Of SCSIdos 3.0

UpdateML

This file is used by the automatic update program to facilitate updating older SCSIdos systems to the new SCSIdos 3.0 format. The UpdateML program uses information from an existing SCSIdos 2.xx MountList to create MountList and DevSetup files that are compatible with SCSIdos 3.0.

Usage: UpdateML <RETURN>

CheckFree

This file is used by the automatic update program to facilitate updating older SCSIdos systems to the new SCSIdos 3.0 format. The program checks for sufficient free disk space.

Usage: CheckFree [device] [Blocks Required] <RETURN>

DefDisk

DefDisk is a public domain program that performs the same function as the AmigaDOS assign command, except that it assigns ALL logical devices to the target device in one operation.

Usage: DefDisk [device] <RETURN>

DevInstall

This is the program that allows the uses user supplied data contained in the DevSetup file to custom configure SCSIdos 3.0 to comply with each user's unique needs. This program should be copied into your 'c' directory and should be called from your s/startup-sequence before MOUNTING any SCSI devices.

Usage: 1> DevInstall @Devs:DevSetup [-QUIET] <RETURN>
Where: -QUIET is optional to suppress screen output.

CLtd.device, SCSI.handler and HardDisk.driver

These files, as described in the section titled *'Overview of SCSIdos 3.0'* earlier in this manual, are the actual operational components of the SCSIdos 3.0 device driver and MUST be in the currently assigned DEVS: directory.

DevSetup

This is the file that contains the data used by DevInstall to configure the system. This text (editable) file allows the user to define all operating parameters and options. There is a tremendous amount of customization possible here, so tread gently. While you can edit this file directly with any text editor, like TxED or CygnusED, there is a more user friendly intuition based 'front-end' called MLed (MountListEditor) supplied that has complete USER HELP functions and will simplify the chore of creating and maintaining this file with its many options.

Mountlist

This is a sample MountList that can be edited (again, use MLed for ease of editing) to comply with your hard drive's surfaces, blocks and high/low cyls and then stored in the DEVS: directory, thereby replacing any old MountLists.

DevSetup.Sample

This is the same HEAVILY commented sample of a DevSetup file that is reproduced in the next section of this documentation. It contains basic documentation on all available options and their impact on the system, along with recommended settings for each.

SCSI-Link.Library

This is the Library that is available for use to all programs wishing to communicate directly with SCSI devices. It is used extensively with the FEdX program as well as other programs on the disk and MUST be present in the Libs: directory of any disk used to Boot a C Ltd. SCSI system.

Auto-Start

This is a script (execute) file that is used during the initialization of a NEW Hard Disk to put the required files and directories onto the Hard Drive.

HardDrive.Startup-Sequence

This is a script (execute) file that is put into the S: directory on a Hard Drive and is then always used during normal startup operations.

Startup-Sequence

This is the initial Startup-Sequence. There are three variations of this file, of which you will only have one! The different variations are supplied with a) Complete Hard Drive systems, b) SCSI Host/Controllers and c) SCSIdos 3.0 Update disks

UserNotes

This is the text file that is type on your screen that reminds you to Back-up your disks..

Floppy.Startup-Sequence

This is a script (execute) file that replaces the Startup-Sequence on your Boot disk and is used as the normal Startup-Sequence during system start-up.

NewLook

This is the C Ltd. Hard Disk Icon. It is copied onto your Hard Drive and renamed as *Disk.info*.

typenote

This is a script (execute) file that causes the text in the UserNotes file to be typed to your screen.

MLed

This is the Intuition-based editor for the MountList and the DevSetup Files.

MLed.info

This is the Icon for the MLed program.

MLed.help

This file contains the 'HELP' screen information used by MLed when the user requests help.

The Utilities/FmtData Directory

This directory contains data files for all of the most commonly used SCSI Device Controller and SCSI Hard Drive that are used with the C Ltd. SCSI Host/Controller. The name of the file will indicate the controller/hard drive to which it applies. The files are in the compressed raw data format that is used by the FEdX program. This is the directory that is searched when the user selects a default type from the FEdX default menu. As additional files are created (by us and by users) they will be made available on our BBS.

The Utilities/HardDrives Directory

This directory contains data files used by the FEdX program during the formatting of a Hard Disk. When the user selects *Load* or *Save* from the FEdX main menu, this directory is automatically searched.

FEdX

This is the Intuition-based editor used in editing the Low-Level formatter program's configuration Files.

FEdX.info

This is the Icon for the MLed program.

CLtd-Format_0, CLtd-Format_1, CLtd-Format_2 and CLtd-Format_3

These are the programs that actually performs the Low-Level format and bad sector substitution.

Inquire.info, Inquire, 1, 1.a and Inquir

These are the Iconx icon and associated script files that perform the SCSI inquire function. This operation queries the SCSI bus and returns information about ALL devices that are properly connected to it.

Check.info, Check, 2, 2.a and SCSIcheck

These are the Iconx icon and associated script files that perform the SCSI Check function. This operation tests the C Ltd. Host with Adaptec SCSI to ST-506 device controllers.

Size.info, Size, 3, 3.a and CAP

These are the Iconx icon and associated script files that perform the SCSI Capacity function. This operation returns the size (in sectors/blocks) of the selected SCSI device connected to the system.

Clear.info, Clear, 4, 4.a and ClearSCSI

These are the Iconx icon and associated script files that perform the Clear SCSI function. This operation resets the SCSI bus and the selected SCSI device connected to the system.

Errors.info, Errors, 5, 5.a and ReadErr

These are the Iconx icon and associated script files that perform a NON-DESTRUCTIVE examination of a SCSI device. The program will verify EVERY sector on your hard drive and report any read errors in both track/surface/byte and sector number format.

Formatter.info, Formatter, 6, 6.a and GenFormat

These are the Iconx icon and associated script files that perform a generic SCSI Low-Level format of any specified SCSI device. This formatter can be used with any SCSI Hard Drive and Most SCSI removable media drives to achieve a quick Low-Level format of the device.

Update_To_3.0 and Update_To_3.0.info

These are the Iconx icon and associated script files that perform the automatic updating of older SCSIdos 2.xx boot disks for use as new SCSIdos 3.0 boot disks.

SCSIdos™ Sample MountList

```
/* Sample MountList for a C Ltd. 20Meg Hard Drive using a MiniScribe 8425s */
/* This example is set up to use the AmigaDOS 1.3 Fast File System         */
/* For additional information about the items in the MountList refer to    */
/* the documentation for AmigaDOS 1.3 (supplied by Commodore-Amiga) and to */
/* the MLed program's Help function for the item in question.              */

DH0:    Device          =  CLtd.device          /* This is Case Sensitive   */
        FileSystem      =  l:FastFileSystem     /* Assumes AmigaDOS 1.3 FFS */
        Unit            =  1000                 /* Ref: DevSetup.Sample     */
        Flags           =  0                    /* Always Use 0             */
        Surfaces        =  4                    /* User defined See Note 1  */
        BlocksPerTrack  =  17                   /* User defined See Note 1  */
        Reserved        =  2                    /* Always Use 2             */
        Interleave      =  0                    /* Always use 0             */
        LowCyl          =  0                    /* User defined See Note 1  */
        HighCyl         =  609                  /* User defined See Note 2  */
        Buffers         =  5                    /* User Optional Value      */
        GlobVec         =  -1                   /* Commodore defined Value  */
        BufMemType      =  3                    /* User Optional Value      */
        Mount           =  1                    /* User defined See Note 3  */
        DosType         =  0x444F5301           /* Commodore defined Value  */
        StackSize       =  4000                 /* User defined See Note 4  */
        #

/*************************************************************************/
/* NOTES:                                                                */
/*                                                                       */
/*      1.  These values are specific to the hard drive mechanism and    */
/*          should correspond to the manufacturer's recommendations.     */
/*                                                                       */
/*      2.  This is the value that is returned by the C Ltd low-level    */
/*          formatter after formatting.                                  */
/*                                                                       */
/*      3.  This value tells the system to mount the drive.              */
/*                                                                       */
/*      4.  Stack size minimum should be 4000                            */
/*                                                                       */
/*************************************************************************/
```

SCSIdos™ Sample DevSetup File

```
/* ---------- Begin Comment Area ----------------------------------------

    File:    Sample Devs Setup File.
    Date:    Aug. 15, 1988

    This is a comment area that is not used by any of the configuration
    software. Comments are delimited with the same characters as "C" source
    files as with the normal AmigaDOS Mountlist.
    In this area I keep track of my system's hardware configuration,
    ie. what unit numbers are assigned to what physical and/or logical devices.
    It is not necessary to have any information here.

    Sample System Configuration

    Unit #      Description
    ------      ----------------------------
    1000        dh0: Adaptec 4000 w/ CMI 6000 20 MEG Fixed Disk
    1010        dh1: Adaptec 4000 w/ CMI 6000 20 MEG Fixed Disk
    1100        dh2: OMTI 3520 w/ Miniscribe 8425 20 MEG Fixed Disk
    1110        dh3: Adaptec 4070 w/ Okidata 526 24 MEG Fixed Disk
    1200        dh4: Seagate 277N 65 MEG Fixed Disk
    2300        KT0: Konica 10 MEG Removable Media (5.25" Floppy)
    2400        db0: Bernoulli LUN 0
    2410        db1: Bernoulli LUN 1
    5000        ---- Laser Printer

------ End of Comment Area ------------------------------------------------ */

    Device CLtd.device              /* Set Device                          */

                                    /* This field tells the DevInstall     */
                                    /* program that the "DEVICE" is in     */
                                    /* a file called CLtd.device.          */
                                    /* NOTE: The file must be in DEVS:     */
                                    /* NOTE: filename IS case sensitive    */

    Define Handler  SCSI.handler    /* Define Handler code file name       */

                                    /* This field tells the "DEVICE"       */
                                    /* that the "HANDLER" for the          */
                                    /* "DEVICE" is in a file called        */
                                    /* SCSI.handler.                       */
                                    /* NOTE: The file must be in DEVS:     */
                                    /* NOTE: filename IS case sensitive    */
```

```
Load Handler      SCSI.handler      /* Load Handler code segment        */

                                    /* Causes the "HANDLER" file (as    */
                                    /* just defined above) to be loaded.*/
                                    /* Execution will not take place    */
                                    /* until the "HANDLER" is needed.   */
                                    /* (ie. a hard drive is mounted)    */

Define Driver     HardDisk.driver   /* Define Driver code file name     */

                                    /* This field tells the "DEVICE"    */
                                    /* that the "DRIVER" for the        */
                                    /* "DEVICE" is in a file called     */
                                    /* HardDisk.driver.                 */
                                    /* NOTE: The file must be in DEVS:  */
                                    /* NOTE: filename IS case sensitive */
                                    /* NOTE: "HardDisk.driver" is used  */
                                    /*       for ALL Hard Drives        */

Load Driver       HardDisk.driver   /* Load Driver code segment         */

                                    /* Causes the "DRIVER" file (as     */
                                    /* just defined above) to be loaded.*/
                                    /* Execution will not take place    */
                                    /* until the "DRIVER" is needed.    */
                                    /* (ie. a hard drive is mounted)    */

Define Driver     LaserPrt.driver   /* Define Driver code file name     */

                                    /* This field tells the "DEVICE"    */
                                    /* that the "DRIVER" for the        */
                                    /* "DEVICE" is in a file called     */
                                    /* LaserPrt.driver.                 */
                                    /* NOTE: The file must be in DEVS:  */
                                    /* NOTE: filename IS case sensitive */
                                    /* NOTE: "LaserPrt.driver" is used  */
                                    /*       for C Ltd.'s LazerXpress   */
                                    /*       SCSI Laser Printer System  */

Load Driver       LaserPrt.driver   /* Load Driver code segment         */

                                    /* Causes the "DRIVER" file (as     */
                                    /* just defined above) to be loaded.*/
                                    /* Execution will not take place    */
                                    /* until the "DRIVER" is needed.    */
                                    /* (ie. you print something!)       */
```

```
Initialize Handler  100             /* Initialize Handler          */

                                    /* This command tells the DevInstall*/
                                    /* program that it should set up    */
                                    /* the "DEVICE" to recognize a      */
                                    /* "HANDLER" with an ID# of 100 and */
                                    /* that "HANDLER" 100 when created  */
                                    /*  will have attributes as provided*/
                                    /*  in the list that follows.       */
                                    /*  NOTE: ID# is an arbitrary value */
                                    /**********************************/
                                    /*  Tech Note: Additional handlers  */
                                    /* may be initialized here for use  */
                                    /* with an Amiga with more that one */
                                    /* C Ltd. SCSI HOST/CONTROLLER.     */
                                    /* In such cases, each handler must */
                                    /* have a unique ID #.              */
                                    /**********************************/
    HandlerCode     SCSI.Handler    /* Code Segment Name           */

                                    /* "HANDLER" ID# 100 will use the   */
                                    /* SCSI.Handler code. (Which was    */
                                    /* loaded above.)                   */

    TaskName        SCSI Bus 1      /* Set Up Task Name            */

                                    /* Assign the task an arbitrary name*/
                                    /* in this case SCSI Bus 1          */

    TaskPriority    15              /* Handler's Task Priority     */

                                    /* This is the priority that should */
                                    /* be assigned to the "HANDLER" when*/
                                    /* it is active.                    */

    StackSize       250             /* Stack Size                  */

                                    /* This is the Stack Size for the   */
                                    /* "HANDLER" task.                  */
                                    /* NOTE: DON'T MESS WITH THIS!!!!!! */
```

```
HostAddress     7           /* Our SCSI address number              */

                            /* This is the SCSI ADDRESS (ie real*/
                            /* SCSI 0-7 numbering system) that  */
                            /* is assigned to the C Ltd. SCSI   */
                            /* HOST/CONTROLLER not the Hard Disk*/
                            /* NOTE: This number can be any  0-7*/
                            /* number that does not conflict    */
                            /* other SCSI devices on the system */
                            /* and is used for Multi-Host app-  */
                            /* lications (aka SCSInet)          */

Flags           NoArbitrate /* Disable Arbitration              */

                            /* This attribute is intended to be */
                            /* used with SCSInet (Multi-Host)   */
                            /* systems to provide full ANSI     */
                            /* arbitration protocol. Options    */
                            /* are "Arbitrate" and "NoArbitrate"*/
                            /* Arbitration MUST be ON on ALL    */
                            /* computers in Multi-Host systems! */
```

```
Flags               NoDisable IRQ      /* Do Not Hold Off Interrupts       */

                                       /* This attribute allows the user to*/
                                       /* select between "Disable" and     */
                                       /* "NoDisable"  When Disable is     */
                                       /* selected, all system interrupts  */
                                       /* will be held off during SCSI bus */
                                       /* data transfer and processed when */
                                       /* SCSI transfer is complete.  This */
                                       /* provides Maximum SPEED at the    */
                                       /* cost of jerky mouse movements    */
                                       /* during hard disk activity and the*/
                                       /* inability to reliably operate    */
                                       /* most telecommunications programs */
                                       /* while Hard Disk activity is in   */
                                       /* progress.  When "NoDisable" is   */
                                       /* selected, all operations will    */
                                       /* act normally, but some minor     */
                                       /* speed degradation (single digit  */
                                       /* percentage - depends totally on  */
                                       /* applications in progress at the  */
                                       /* time) will may occur with some   */
                                       /* hard drives.                     */
                                       /* This is a GLOBAL setting for all */
                                       /* units using this "HANDLER" If it */
                                       /* is set to "NoDisable" it can be  */
                                       /* over-ridden at the unit level.   */
                                       /* If "Disable" is set, it can't.   */

ExitInit

                                       /* This command ends the attribute  */
                                       /* list and the initialization of   */
                                       /* "HANDLER" ID# 100                */
```

```
/*****************************************************************************/
/*
     The following section describes the initialization of a Logical device
     (ie. Hard Drive partition) and its assigned attributes.

     For systems with multiple Hard Disks and/or several partitions on a
     Hard Disk, one of these Initializations is required for each actual
     physical Hard Drive Unit, or for each partition on a single Hard
     Drive if the user desires to assign different attributes (such as
     software WriteProtect ie. see below.) to each partition.

*/

Initialize Unit     1000                /* Initialize Unit                  */

                                        /* This command tells the DevInstall*/
                                        /* program that it should set up    */
                                        /* the "DEVICE" to recognize a      */
                                        /* "DRIVER" with an ID# of 1000 and */
                                        /* that "HANDLER" 1000 when created */
                                        /*  will have attributes as provided*/
                                        /*  in the list that follows.       */
                                        /*  NOTE: ID# is an arbitrary value */
                                        /************************************/
                                        /*  Tech Note: Additional drivers   */
                                        /* may be initialized here for use  */
                                        /* with an Amiga with more that one */
                                        /* Hard Drive Unit and/or partition */
                                        /* as described above in the header */
                                        /* for this section.                */
                                        /************************************/

     DriverCode     HardDisk.driver     /* Unit driver code segment         */

                                        /* "UNIT" ID# 1000 will use the     */
                                        /* HardDisk.driver code. (Which was */
                                        /* loaded above.)                   */

     HandlerNumber  100                 /* Uses SCSI From Above         */

                                        /* This tells "UNIT" ID# 1000 that  */
                                        /* it talks through "HANDLER" 100   */
```

```
TaskName        Unit 1000       /* Set Up Task Name                */

                                /* Assign the task an arbitrary name*/
                                /* in this case Unit 1000          */

TaskPriority    12              /* Unit's Task Priority            */

                                /* This is the priority that should */
                                /* be assigned to the UNIT's task   */
                                /* when it is active.              */

StackSize       250             /* Stack Size                      */

                                /* This is the Stack Size for the   */
                                /* UNIT's task.                    */
                                /* NOTE: DON'T MESS WITH THIS!!!!!! */

AutoRetries     4               /* Number Of Auto Retries          */

                                /* RANGE 0 to 255                  */
                                /* This specifies the number of    */
                                /* retries that will be made       */
                                /* automatically by the driver     */
                                /* before reporting to AmigaDOS    */
                                /* that there has been a Read/Write */
                                /* error and that the user should  */
                                /* be shown the dreaded            */
                                /* Read/Write Error Requester.     */

BufferSize      7               /* Number Of Block Buffers To Use  */

                                /* RANGE 0 to 255                  */
                                /* The number of 512 byte buffers  */
                                /* can be specified here. The user  */
                                /* can tweak this value, but we    */
                                /* recommend 7 for now.            */

UnitAddress     0               /* Controller's SCSI Address       */

                                /* This refers to the ACTUAL SCSI   */
                                /* UNIT NUMBER of the Hard Drive or */
                                /* SCSI device controller (ie. OMTI */
                                /* or Adaptec) that this logical   */
                                /* device is on.                   */
```

```
Unit LUN          0          /* Logical Unit Number / Controller */

                             /* This refers to the Logical Unit  */
                             /* Number (NOT TO BE CONFUSED WITH  */
                             /* A LOGICAL DEVICE) of a physical  */
                             /* device which is addressed as one */
                             /* of several devices controlled by */
                             /* a SCSI device controller (above) */
                             /* which is capable of controlling  */
                             /* multiple devices.                */

Flags             NoDisable IRQ  /* Hold Of Interrupts           */

                             /* This attribute allows the user to*/
                             /* select between "Disable" and     */
                             /* "NoDisable"  When Disable is     */
                             /* selected, all system interrupts  */
                             /* will be held off during SCSI bus */
                             /* data transfer and processed when */
                             /* SCSI transfer is complete.  This */
                             /* provides Maximum SPEED at the    */
                             /* cost of jerky mouse movements    */
                             /* during hard disk activity and the*/
                             /* inability to reliably operate    */
                             /* most telecommunications programs */
                             /* while Hard Disk activity is in   */
                             /* progress.  When "NoDisable" is   */
                             /* selected, all operations will    */
                             /* act normally, but some minor     */
                             /* speed degradation (single digit  */
                             /* percentage - depends totally on  */
                             /* applications in progress at the  */
                             /* time) will may occur with some   */
                             /* hard drives.                     */
                             /* Selecting the "Disable" here will*/
                             /* over-ride the GLOBAL selection   */
                             /* of "NoDisable" made in the       */
                             /* "HANDLER" Initialization.        */
                             /* If the GLOBAL selection is set   */
                             /* to "Disable" it can't be over-   */
                             /* ridden by any setting here       */
```

```
Flags           NoDisableBuffer   /* Disable Buffering                  */

                                  /* This allows the user to dis-       */
                                  /* able any and all buffering. When   */
                                  /* "Disable" is selected, all data    */
                                  /* will be sent directly to the       */
                                  /* memory area specified by the       */
                                  /* requesting program.                */
                                  /************************************/
                                  /* Tech Note: This option is pro-     */
                                  /* for custom applications software   */
                                  /* designed to communicate directly   */
                                  /* with the disk drive. General use   */
                                  /* of this option is discouraged.     */
                                  /************************************/

Flags           BlockRead         /* Block Reads                        */

                                  /* Makes things a lot faster if       */
                                  /* "BlockRead" is enabled.  "Block-   */
                                  /* Read" works with everything we     */
                                  /* have tested EXCEPT Adaptec 4070    */
                                  /* RLL controllers.  If you have an   */
                                  /* Adaptec 4070 you MUST select the   */
                                  /* "NoBlockRead" option here. But,    */
                                  /* the Adaptec 4000 (MFM) controller  */
                                  /* will work with "BlockRead" ON.     */

Flags           NoBlockWrite      /* Block Writes                       */

                                  /* This option is currently not im-   */
                                  /* plemented, set to NoBlockWrite.    */

Flags           NoSectorSize256   /* 256 Byte Sectors                   */

                                  /* Normally set to NoSectorSize256,   */
                                  /* you can set 'SectorSize256" for    */
                                  /* use with devices that require      */
                                  /* 265 byte blocks like the IOmega    */
                                  /* 20 Meg Bernoulli Box.              */
```

```
    Flags           AdaptiveBuffer    /* Adaptive Buffering              */

                                      /* This flag controls our unique   */
                                      /* and highly proprietary dynamic  */
                                      /* buffering system which optimizes */
                                      /* buffer size to current usage.   */
                                      /* The buffer upper limit is set   */
                                      /* by BufferSize (above) except for */
                                      /* very unique situations.         */
                                      /* "NoAdaptiveBuffer" disables this */
                                      /* option and should only be used  */
                                      /* for diagnostic purposes.        */

    Flags           NoWriteProtect    /* Write Protect                   */

                                      /* When set to "WriteProtect", this */
                                      /* option allows you to software   */
                                      /* Write Protect all Logical Units */
                                      /* using the Unit # of 1000. If you */
                                      /* need to write protect one or    */
                                      /* several partitions on a Drive,  */
                                      /* but wish to leave one or several */
                                      /* other partitions un-protected,  */
                                      /* you MUST set-up another Initial- */
                                      /* ize Unit entry with a different */
                                      /* Unit ID# for those Logical Units */
                                      /* (ie. partitions, devices, etc.) */

ExitInit

                                      /* This command ends the attribute */
                                      /* list and the initialization of  */
                                      /* UNIT ID # 1000                  */
```

```
/*****************************************************************************/
/*****************************************************************************/
/*     The following are examples of other UNIT initialization entries       */
/*****************************************************************************/
/*****************************************************************************/


Initialize Unit     1010            /* Initialize Unit                       */
     DriverCode     HardDisk.driver /* Unit driver code segment              */
     HandlerNumber  100             /* Uses SCSI From Above                  */
     TaskName       Unit 1010       /* Set Up Task Name                      */
     TaskPriority   12              /* Unit's Task Priority                  */
     StackSize      250             /* Stack Size                            */
     AutoRetries    4               /* Number Of Auto Retries                */
     BufferSize     7               /* Number Of Block Buffers To Use        */
     UnitAddress    0               /* Controller's SCSI Address             */
     Unit LUN       1               /* Logical Unit Number / Controller      */
     Flags          NoDisable IRQ   /* Don't Hold Of Interrupts              */
     Flags          NoDisableBuffer /* Don't Disable Buffering               */
     Flags          BlockRead       /* Do Block Reads                        */
     Flags          NoBlockWrite    /* Don't Do Block Writes                 */
     Flags          NoSectorSize256 /* 512 Byte Sectors                      */
     Flags          AdaptiveBuffer  /* Use Adaptive Buffering                */
     Flags          NoWriteProtect  /* Don't Write Protect                   */
ExitInit


Initialize Unit     1100            /* Initialize Unit                       */
     DriverCode     HardDisk.driver /* Unit driver code segment              */
     HandlerNumber  100             /* Uses SCSI From Above                  */
     TaskName       Unit 1100       /* Set Up Task Name                      */
     TaskPriority   12              /* Unit's Task Priority                  */
     StackSize      250             /* Stack Size                            */
     AutoRetries    4               /* Number Of Auto Retries                */
     BufferSize     7               /* Number Of Block Buffers To Use        */
     UnitAddress    1               /* Controller's SCSI Address             */
     Unit LUN       0               /* Logical Unit Number / Controller      */
     Flags          NoDisable IRQ   /* Don't Hold Of Interrupts              */
     Flags          NoDisableBuffer /* Don't Disable Buffering               */
     Flags          BlockRead       /* Do Block Reads                        */
     Flags          NoBlockWrite    /* Don't Do Block Writes                 */
     Flags          NoSectorSize256 /* 512 Byte Sectors                      */
     Flags          AdaptiveBuffer  /* Use Adaptive Buffering                */
     Flags          NoWriteProtect  /* Write Protect                         */
ExitInit
```

```
Initialize Unit      1110                /* Initialize Unit                    */
     DriverCode      HardDisk.driver     /* Unit driver code segment           */
     HandlerNumber   100                 /* Uses SCSI From Above               */
     TaskName        Unit 1110           /* Set Up Task Name                   */
     TaskPriority    12                  /* Unit's Task Priority               */
     StackSize       250                 /* Stack Size                         */
     AutoRetries     4                   /* Number Of Auto Retries             */
     BufferSize      7                   /* Number Of Block Buffers To Use     */
     UnitAddress     1                   /* Controller's SCSI Address          */
     Unit LUN        1                   /* Logical Unit Number / Controller   */
     Flags           Disable IRQ         /* Do Hold Of Interrupts              */
     Flags           DisableBuffer       /* Do Disable Buffering               */
     Flags           NoBlockRead         /* Don't Do Block Reads               */
     Flags           NoBlockWrite        /* Don't Do Block Writes              */
     Flags           NoSectorSize256     /* 512 Byte Sectors                   */
     Flags           AdaptiveBuffer      /* Use Adaptive Buffering             */
     Flags           NoWriteProtect      /* Don't Write Protect                */
ExitInit


Initialize Unit      1200                /* Initialize Unit                    */
     DriverCode      HardDisk.driver     /* Unit driver code segment           */
     HandlerNumber   100                 /* Uses SCSI From Above               */
     TaskName        Unit 1200           /* Set Up Task Name                   */
     TaskPriority    12                  /* Unit's Task Priority               */
     StackSize       250                 /* Stack Size                         */
     AutoRetries     4                   /* Number Of Auto Retries             */
     BufferSize      7                   /* Number Of Block Buffers To Use     */
     UnitAddress     2                   /* Controller's SCSI Address          */
     Unit LUN        0                   /* Logical Unit Number / Controller   */
     Flags           NoDisable IRQ       /* Don't Hold Of Interrupts           */
     Flags           NoDisableBuffer     /* Don't Disable Buffering            */
     Flags           BlockRead           /* Do Block Reads                     */
     Flags           NoBlockWrite        /* Don't Do Block Writes              */
     Flags           NoSectorSize256     /* 512 Byte Sectors                   */
     Flags           AdaptiveBuffer      /* Use Adaptive Buffering             */
     Flags           WriteProtect        /* Do Write Protect                   */
ExitInit
```

```
Initialize Unit     2300                /* Initialize Unit                    */
     DriverCode      HardDisk.driver    /* Unit driver code segment           */
     HandlerNumber   100                /* Uses SCSI From Above               */
     TaskName        Unit 2300          /* Set Up Task Name                   */
     TaskPriority    12                 /* Unit's Task Priority               */
     StackSize       250                /* Stack Size                         */
     AutoRetries     4                  /* Number Of Auto Retries             */
     BufferSize      7                  /* Number Of Block Buffers To Use     */
     UnitAddress     3                  /* Controller's SCSI Address          */
     Unit LUN        0                  /* Logical Unit Number / Controller   */
     Flags           NoDisable IRQ      /* Don't Hold Of Interrupts           */
     Flags           NoDisableBuffer    /* Don't Disable Buffering            */
     Flags           BlockRead          /* Do Block Reads                     */
     Flags           NoBlockWrite       /* Don't Do Block Writes              */
     Flags           NoSectorSize256    /* 512 Byte Sectors                   */
     Flags           AdaptiveBuffer     /* Use Adaptive Buffering             */
     Flags           NoWriteProtect     /* Don't Write Protect                */
ExitInit
```

```
Initialize Unit     2400                /* Initialize Unit                    */
     DriverCode      HardDisk.driver    /* Unit driver code segment           */
     HandlerNumber   100                /* Uses SCSI From Above               */
     TaskName        Unit 2400          /* Set Up Task Name                   */
     TaskPriority    12                 /* Unit's Task Priority               */
     StackSize       250                /* Stack Size                         */
     AutoRetries     4                  /* Number Of Auto Retries             */
     BufferSize      7                  /* Number Of Block Buffers To Use     */
     UnitAddress     4                  /* Controller's SCSI Address          */
     Unit LUN        0                  /* Logical Unit Number / Controller   */
     Flags           NoDisable IRQ      /* Don't Hold Of Interrupts           */
     Flags           NoDisableBuffer    /* Don't Disable Buffering            */
     Flags           BlockRead          /* Do Block Reads                     */
     Flags           NoBlockWrite       /* Don't Do Block Writes              */
     Flags           SectorSize256      /* 256 Byte Sectors                   */
     Flags           AdaptiveBuffer     /* Use Adaptive Buffering             */
     Flags           NoWriteProtect     /* Don't Write Protect                */
ExitInit
```

```
Initialize Unit      2410                /* Initialize Unit                    */
    DriverCode       HardDisk.driver     /* Unit driver code segment           */
    HandlerNumber    100                 /* Uses SCSI From Above               */
    TaskName         Unit 2410           /* Set Up Task Name                   */
    TaskPriority     12                  /* Unit's Task Priority               */
    StackSize        250                 /* Stack Size                         */
    AutoRetries      4                   /* Number Of Auto Retries             */
    BufferSize       7                   /* Number Of Block Buffers To Use     */
    UnitAddress      4                   /* Controller's SCSI Address          */
    Unit LUN         1                   /* Logical Unit Number / Controller   */
    Flags            NoDisable IRQ       /* Don't Hold Of Interrupts           */
    Flags            NoDisableBuffer     /* Don't Disable Buffering            */
    Flags            BlockRead           /* Do Block Reads                     */
    Flags            NoBlockWrite        /* Don't Do Block Writes              */
    Flags            SectorSize256       /* 256 Byte Sectors                   */
    Flags            AdaptiveBuffer      /* Use Adaptive Buffering             */
    Flags            NoWriteProtect      /* Don't Write Protect                */
ExitInit


/* Updated Laser Printer DevSetup file entry    */

Initialize Unit      5000                /* Initialize Unit                    */
    DriverCode       LaserPrt.driver     /* Unit driver code segment           */
    HandlerNumber    100                 /* Uses SCSI From Above               */
    TaskName         LaserPrinter        /* Set Up Task Name                   */
    TaskPriority     10                  /* Unit's Task Priority               */
    StackSize        250                 /* Stack Size                         */
    UnitAddress      3                   /* Printers SCSI Address              */
    MarkerHeight     200                 /* Marker Height          (20 - 200)  */
    MarkerX          0                   /* Marker X Position      (0 - 320)   */
    MarkerY          0                   /* Marker Y Position      (0 - 200)   */
    MarkerType       0                   /* Marker Type            (0 - 1)     */
ExitInit


Exit
```

Setting the SCSI Unit Address.

Every SCSI device (ie. Hard Drives, CD ROMs, Optical Drives, Laser Printers etc.) must have a device number. The instructions supplied with each device will explain how to set or change the device number of that particular product. (It is usually done by means of setting a dip switch or moving a jumper.) The C Ltd. SCSI Host/Controller will recognize SCSI devices numbered zero (0) through seven (7). These device numbers correspond to the UNIT numbers as used in the DevSetup File for the SCSI device. Most SCSI devices are set at the factory to respond as device number zero. Your C Ltd. Hard Drive is set up as device 0, be sure to change the device numbers of any additional SCSI devices so that you have all of the SCSI devices matching their corresponding MountLists, all are arranged in ascending order starting with device Zero (0), and be sure that there are no duplications of device numbers.

There are several different styles of Hard Drives and Hard Drive enclosures available, to change the device number of most C Ltd. Hard Drives follow the directions here. Generally, SCSI device types other than the type discussed here will include additional documentation outlining the proper procedures for changing the device's SCSI address, but if the SCSI device you have doesn't seem to conform to the instructions supplied here, please call our Customer Service line for assistance.

1. **Turn off all power to the Amiga and the Hard Drive.**

2. Turn over your Hard Drive and locate the hole that would seem to be located in an unusual position, slightly off center, but about in the middle of the drive. Locate the exposed set of jumper pins located on the circuit board seen through the hole.

3. Use jumpers to set the desired device number as follows:

Device #	Pins A-B	Pins C-D	Pins E-F
0	OPEN	OPEN	OPEN
1	SHORT	OPEN	OPEN
2	OPEN	SHORT	OPEN
3	SHORT	SHORT	OPEN
4	OPEN	OPEN	SHORT
5	SHORT	OPEN	SHORT
6	OPEN	SHORT	SHORT
7	SHORT	SHORT	SHORT

4. Now , be sure that your MountList and/or DevSetup files reflects the changes you have just made and power-up your system.

Connecting Additional SCSI Devices

Before doing any work, read this section completely, then please follow basic precautions that should be taken when dealing with any electronic circuitry as outlined on page 4 of the instruction manual.

There are actually two similar systems defined for SCSI network wiring and device interconnection. One is the normally accepted SCSI standard (defined by ANSI - the American National Standards Institute) as used by most manufacturers of SCSI devices, and the other is a pseudo-standard as used by Apple computers on its Macintosh computers. Your C Ltd. Hard Drive and C Ltd. SCSI Host/Controller provide the proper connections for both systems. Your C Ltd. SCSI Host/Controller and Hard Drive will be delivered configured with the Apple/MacIntosh system (described below) unless it was originally special ordered otherwise.

The specifications for adding units to the SCSI network are defined in the Standards for Small Computer Systems Interfacing. These standards are very specific and should be followed exactly. (Most of the time! [Ed])

Basically, a properly laid out SCSI system will look like one long cable starting at the SCSI Host/Controller (the C Ltd. SCSI Card in this case) and ending at the last SCSI device, with up to six tap-offs in the middle that connect to the various SCSI devices tied into the system. The standards also specify that there should be proper termination of the lines at both ends. Both the SCSI standard system and the Apple Macintosh pseudo-system follow these general rules, but differ in the type of cables and connectors used.

Most probably your system is configured using the Apple Macintosh pseudo-standard. With this system, the SCSI Host (the C Ltd. SCSI Host/Controller) is fitted with a 25 pin connector. A round shielded 25 conductor cable then connects the SCSI Host to a 50 pin connector at the SCSI device. (This is the cable that is supplied with your C Ltd. Hard Drive.) Each C Ltd. SCSI device has two 50 pin connectors connected in parallel, so it doesn't matter which one is used. Additional SCSI devices are simply connected to the first device in 'daisy chain' fashion by using a round shielded 25 conductor cable with the appropriate 50 pin connector on each end. (C Ltd. maintains a stock of these cables, so if your dealer does not have what you need, call our Customer Service Department at (316) 267-0202 to order additional cables.)

Regarding the standard for line termination, the C Ltd. SCSI Controller is the Host and is therefore usually at one end of the line, it has the proper termination built into it. With your present single unit system, the other end of the line is the C Ltd. Hard Drive, so it is shipped properly terminated. When you add another device you may put it between the C Ltd. SCSI Controller and the C Ltd. SCSI Hard Drive. In such a case, if the device to be added has termination resistors installed, they should be removed. How to remove the terminating resistor packs should by explained in the documentation provided with your SCSI device, as it is not the same for all units. (It usually consists of removing one, two or three socketed resistor packs, and/or moving a jumper or two.)

If you add another device and put it after the C Ltd. SCSI Hard Drive, you must remove the terminating resistor packs on the C Ltd. Hard Drive. The termination scheme is different for each of the various sizes of the C Ltd. Hard Drives, so if you wish to remove the terminating resistor packs you should call our Customer Service Department at (316) 276-0202 for instructions.

As for the other SCSI wiring method, it is based on using a 50 conductor ribbon (or flat) cable, connected to the SCSI Host (the C Ltd. SCSI Host/Controller) and each SCSI device through a 50 pin header on the circuit board of each SCSI device that consists of two rows of 25 pins each, and matching 50 hole connectors attached to the ribbon cable. The C Ltd. SCSI Host/Controller has the required 50 pin connector located on its circuit board. The connector is accessible by removing the Host/Controller's cover. Some SCSI devices provide two sets of 50 pin male connectors at the device. If this is the case, simply connect a 50 conductor cable from the C Ltd. SCSI Controller to one of the 50 pin male connectors on your new drive, then connect the 50 conductor cable attached to the C Ltd. Hard Drive to the other. If your drive has only one 50 pin male connector on the drive, you will have to obtain a "T" connector that will provide the necessary dual connections. Alternatively, you can have a special 50 conductor cable made that has a "tap-off" connector in the middle to connect to your new SCSI device. Of the two options the latter is the preferred method, it will also allow you to specify the length of the cable and the distance between devices, so as to eliminate any extra cable. Though SCSI standards allow distances of up to 30 feet between units, unnecessary cable should be eliminated as it is a potential source of problems. A suitable cable and/or the correct "T" adapter should be easily located at any electronics store, or may be purchased from C Ltd. (Call Customer Service at (316) 267-0202.)

It is also possible to create combinations of these two systems by using both the 25 Pin connector and the 50 pin connector on the C Ltd. SCSI host/controller. If you try to use a combination like this the C Ltd. SCSI host/controller will no longer be at the end of the line, so its termination resistors must be removed.

Once you are sure that you have the correct cable and connector always REMEMBER to observe the cautions listed earlier. If you are going to be using the round shielded cable method of wiring, just plug the cables in where they fit as described above. If, however, you are going to use the flat ribbon cable wiring system, the 50 pin connector is inside the C Ltd. SCSI controller case, so remove the screws holding the C Ltd. SCSI Controller's case together, remove the cover, uncovering the circuit board. Locate the Male 50 Pin connector on the circuit board. Orient the circuit board so that the side of the circuit board that the components are mounted to is facing you, and the 25 pin and 50 pin connectors are to your right. Locate the pin on the 50 pin connector that is marked as "1" and locate the edge of the 50 conductor ribbon cable that is marked with the black or red stripe. Connect the cable to the board with the marked edge of the cable closest to pin "1" of the connector and route the cable so it passes over the 25 pin connector. Now you must re-assemble the C Ltd. SCSI Host/Controller, passing the 50 conductor ribbon cable out of the case through the slot provided at the rear of the case.

At the Hard Drive end of the line, all C Ltd. SCSI Hard Drives are supplied with two Apple Macintosh type 50 pin connectors mounted on the back of the drive. If you are using the round shielded cable wiring method, these connectors will fit the connectors on the end of the cable. (The two connectors are connected in parallel, so either connector can be used as an input or an output.)

If you are using the 50 conductor flat ribbon cable, the matching 50 pin connector to plug on your cable is inside the hard drive. REMEMBER to turn off your hard drive and observe the cautions addressed earlier in this manual, then:

1. Take the cover off your Hard Drive and trace the cable from the two 50 pin connectors on the back of the case back to where it connects to the Hard Drive Unit. There you should find the cable is connected to the correct type of 50 Pin male connector on the circuit board.

2. Unplug the connector on the cable coming from the connectors on the rear panel of the enclosure, making sure to note the orientation (pin 1) of the cable and connector.

3. You can now use the 50 pin connector for the installation as described above.

Extending The SCSI/1000 Host/Controller

The APTB Pass Thru Board for the Amiga A-1000

If your pocketbook permits the use of more than one expansion device on the Amiga expansion bus, you will require the use of the SCSI Card's available pass-thru option and you should order P/N APTB ($19.95) from your dealer or call C Ltd.'s Customer Service at (316) 267-0202 for assistance.

We chose to use this method of extending the Amiga expansion bus first, because (from previous experience with our aMEGA 1 MEG RAM Board) we found it to be a much more reliable and durable mechanical assembly, especially in situations where the printer (Daisy Wheel printers are the worst) is on the same stand or desk as the computer and the vibration can be transmitted to the connector assembly. And second because not everyone will need the expansion capability, so only those who will use it will have to pay for it.

Installing the APTB Adapter:

1. Be sure to follow the basic precautions as outlined on Page 4 of the manual.
2. **Be sure that all power is off to your computer and any devices attached to it.**
3. If the SCSI/1000 Card is plugged into your Amiga A-1000 unplug it from the computer.
4. Locate and inspect the pass-thru connector cover. The cover is located directly opposite the exposed connector that plugs into the expansion port of your Amiga A-1000.
5. Remove the two screws that hold the cover in place. This will expose the pass-thru connector located under the cover.
6. Inspect the pass-thru board (P/N APTB) and locate the side of the board marked TOP and FRONT.
7. With the TOP of the pass-thru board UP and the FRONT of the board to the FRONT, firmly press the pass-thru board into the pass-thru connector on the C Ltd. SCSI Host/Controller.
8. Replace the C Ltd. SCSI Host/Controller as per the instructions at the beginning of this manual.
9. Install additional devices as per the instructions provided by the manufacturer.

Backing up Your Hard Drive.

If you bought a complete C Ltd. Hard Drive system, you may have noticed that there is a substantial amount of Public Domain software provided on it. On your Hard Drive you will find at least one and maybe more, Public Domain (PD) Back-up Utility. We are constantly updating and replacing the Back-up Utilities, so complete instructions here would be out-dated by the time you read them. Refer to the ReadMe files in the Docs on the Hard Drive for the most recent version of the back-up that we are providing and for the proper documentation.

In addition to these PD programs, we have tested (and most of use) the *'QuarterBack'* Back-up Utility from Central Coast Software. This Back-up Utility is simply a pleasure to use and provides a very quick and reliable back-up of your data.

Back-up of your data is very important, though we are often able to recover data from drives that have crashed, it is not always possible to do so. Reliable back-up systems are available from C Ltd. in four forms:

1. A 10 Meg Removable Media Back-up Drive - $899.95
   A 44 Meg Removable Media Back-up Drive - $1299.95
2. A 60 Meg Tape Back-up System - $699.95
3. An 800 Meg WORM Optical Drive System - $3999.95
4. Another Hard Drive. (Call for prices)

C Ltd. offers Hard Drives in all sizes upto and including 1.2 Gigabytes.

All of the above devices are available now from C Ltd., any of these devices will plug into your existing system and use the software you now have to provide fast, safe back-up of your data. An image backup of a partition to any of the above devices requires less than 15 seconds per Megabyte.

Partitioning Your Hard Drive

OR

Two Drives Are Better Than One.

The software supplied with your C Ltd. SCSI Controller supports full disk partitioning under Amiga DOS 1.3. You can therefore set up one or more hard drive as if it were actually two (or more) complete hard drives. Each drive will have a different DOS device number (ie. DH0:, DH1:, DH2:, etc.), a different volume name (for example: Main, CSources, Back-ups or whatever) and its own format and directories. Each partition is a device, not a physical device like the Hard Drive itself, but what the Amiga refers to as a *'Logical Device'* meaning it exists in logic but not in reality. There are several advantages to partitioning your drive into two or more logical devices. For one thing, AmigaDOS will find things faster if it has less directories and sub-directories to look through and it will also be faster because it will have less physical distance to move the Hard Drive's read head when searching for a track, so splitting up your files onto several logical devices (partitions) will make things run a bit faster.

Another reason to partition Hard Drives is that each partition can be AmigaDOS formatted independently of the other partitions on the drive, just as if each was a totally separate Hard Drive. Using this system to store important files in several partitions on your hard drive is an inexpensive and efficient back-up alternative. (**NOTE:** Because the Low-Level format can only be performed on the entire drive, this back-up method should not be considered as totally secure.) If, for example you developed an unrecoverable error of some type, on one of the drive's partitions, you could still read the files off of the other partitions. This would allow you to re-format just the offending partition, and use the files in the other undamaged partition(s) to rebuild the newly re-formatted partition. Under this system, the only time you would be in danger of losing all of your data would be in the case of a catastrophic hard drive failure, such as a motor or circuit board failure.

NOTE: We still recommend that you retain backups of your important files on floppy disks or tape, but disk partitioning will even make those back-up operations easier and faster.

IMPORTANT NOTE: Partitioning a hard drive requires re-formatting of the Hard Drive which will erase ALL data from the Hard Drive, so be sure you have made a back-up of all important data before you begin.

All you need to do to partition your disk is modify the MountList and use AmigaDOS Format command to format the desired partitions. On the following pages is an example of a MountList used to partition a Hard Drive, followed by the appropriate mounting and formatting commands that will partition a 20 MEG drive into four partitions, the example file has remarks in it where modifications have been made.

```
/*  An example MOUNTLIST file enabling a C Ltd. AM-24  24 MEG  */
/*  Hard Disk to be partitioned into four (4) partitions       */
/*  to be mounted as DH0:, DH1:, DH2:, and DH3:                */


DH0:        Device        = CLtd.device
            FileSystem    = L:FastFileSystem
            Unit          = 1000
            Flags         = 0
            Surfaces      = 4
            BlocksPerTrack = 18
            Reserved      = 2
            Interleave    = 0
            LowCyl        = 0
            HighCyl       = 159  /* New HighCyl */
            Buffers       = 5
            BufMemType    = 3
            GlobVec       = 1
            Mount         = 1
            DosType       = 0x444F5301
            StackSize     = 4000
            #
DH1:        Device        = CLtd.device
            FileSystem    = L:FastFileSystem
            Unit          = 1000
            Flags         = 0
            Surfaces      = 4
            BlocksPerTrack = 18
            Reserved      = 2
            Interleave    = 0
            LowCyl        = 160  /* LowCyl = DH0:'s HighCyl + 1 */
            HighCyl       = 319  /* New HighCyl */
            Buffers       = 5
            BufMemType    = 3
            GlobVec       = 1
            Mount         = 1
            DosType       = 0x444F5301
            StackSize     = 4000
            #
```

```
DH2:        Device          = CLtd.device
            FileSystem      = L:FastFileSystem
            Unit            = 1000
            Flags           = 0
            Surfaces        = 4
            BlocksPerTrack  = 18
            Reserved        = 2
            Interleave      = 0
            LowCyl          = 320  /* LowCyl = DH1:'s HighCyl + 1 */
            HighCyl         = 479  /* New HighCyl */
            Buffers         = 5
            BufMemType      = 3
            GlobVec         = 1
            Mount           = 1
            DosType         = 0x444F5301
            StackSize       = 4000
            #

DH3:        Device          = CLtd.device
            FileSystem      = L:FastFileSystem
            Unit            = 1000
            Flags           = 0
            Surfaces        = 4
            BlocksPerTrack  = 18
            Reserved        = 2
            Interleave      = 0
            LowCyl          = 480  /* LowCyl = DH2:'s HighCyl + 1 */
            HighCyl         = 647  /* Original HighCyl */
            Buffers         = 5
            BufMemType      = 3
            GlobVec         = 1
            Mount           = 1
            DosType         = 0x444F5301
            StackSize       = 4000
            #
```

Please notice that in the above file, the only things that changed were the Device (DH0:, DH1:, DH2: and DH3:) and the LowCyl and HighCyl Values. So, all you really need to do to partition a drive under Amiga DOS 1.3 is assign the drive's cylinders to different device numbers. Also notice that the partitions do not need to be the same size, but the LowCyl of one partition is ***always one higher*** than the HighCyl of the previous partition.

Partitions can be of any size. If partitions are the same size, however, you can use the AmigaDOS DiskCopy command to quickly back up one partition to another. It is also possible to set up a partition that is the same size as an Amiga 3.5" floppy disk. You could then use the DiskCopy command to quickly make duplicate copies of floppy disks, or load all of the information from a floppy disk onto your Hard Drive.

After the drives are installed, you can use the AmigaDos Format program to format the individual partitions of your drive. (Format with the QUICK option takes about .7 seconds for any size partition, the full AmigaDOS Format takes about 40 seconds per megabyte.) With the AmigaDOS format program, for example you would type:

1> DF0:System/Format drive DH0: Name Main <Return>

1> DF0:System/Format drive DH1: Name Back-Ups <Return>

1> DF0:System/Format drive DH2: Name CSources <Return>

1> DF0:System/Format drive DH3: Name M2Sources <Return>

Once you have the MountList setup properly to reflect the way you want to partition your drive, and have formatted the drives, you must install the devices using the Mount command. (the Mount command could either be issued automatically during the startup-sequence by altering the s/startup-sequence of your Boot disk, or any of the partitions could be mounted individually by issuing the appropriate Mount command from any CLI command line.) For Example:

1> Mount DH0: <Return>

1> Mount DH1: <Return>

1> Mount DH2: <Return>

1> Mount DH3: <Return>

SPECIAL NOTE: The Amiga Workbench screen will not display an icon for any device or partition added to your system (even though it has been properly mounted) until after the first time that the device or partition is actually accessed. Any command that addresses the device or partition may be used to trigger the icon, for example:

1> DiskChange DH1: <RETURN>

1> CD DH1: <RETURN>

1> Dir DH1: <RETURN>

1> Run DH1:Whatever <RETURN>

Any of the above commands (or any similar commands) will be sufficient to cause the Amiga WorkBench to display the icon for your device or partition on the WorkBench screen.

REMEMBER: If one of the partitions should become corrupted the data on the other three would still be available. The damaged partition could be re-formatted (using the AmigaDOS format command) without affecting any of the information stored in any other partitions.

You can determine the size of each partition with the following:

Partition Size = NumberOfCylinders
* NumberOfSurfaces
* BlocksPerTrack
* BytesPerBlock

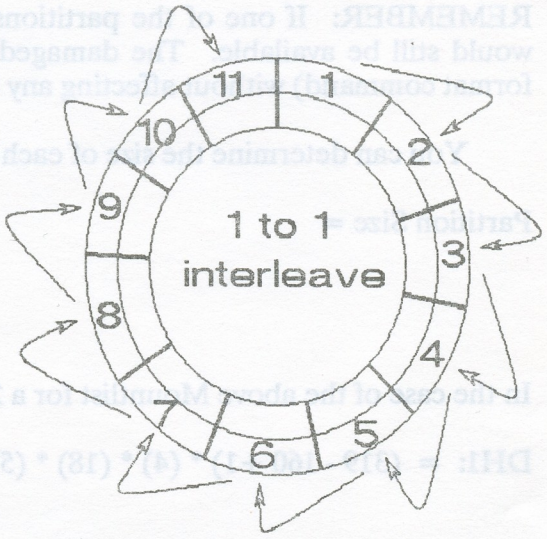
In the case of the above Mountlist for a 24Meg Hard Drive, for example:

DH1: = (319 - 160 +1) * (4) * (18) * (512) = 5898240 Bytes Free (5.898 Megs)

The Interleave Factor

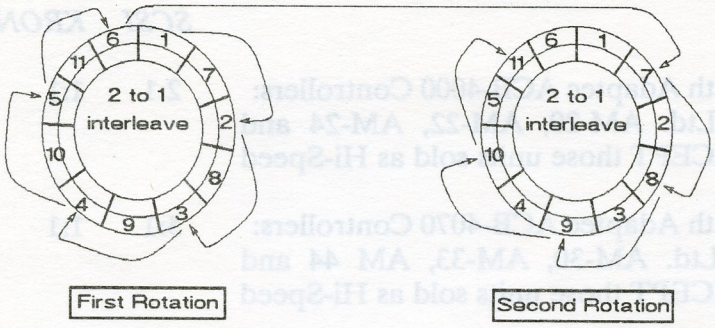
The term INTERLEAVE refers to a method of organizing groups of data on the physical surface of a disk. All disks (Hard, Floppy, Optical, etc.) store data on their surfaces in concentric rings called TRACKs. Most storage devices have two or more SURFACES on which data is stored, some devices may have 16 or more SURFACES. The set of all the TRACKSs in the same position on all the SURFACES is called a CYLINDER. If a storage device has two storage surfaces, then each CYLINDER would consist of two tracks, four SURFACES, four tracks per CYLINDER, etc. There may be as few as 40 CYLINDERs on a disk, as in the case of the Commodore 5.25" Floppy Disk, or as many as 2000 or more, as in the case of the C Ltd. 800 Megabyte Optical Disk. Each of the TRACKSs is further divided into groups of data called BLOCKs. As with CYLINDERs, there may be as few as 11 BLOCKs (on some drives even less) on each CYLINDER, as in the case of the Amiga 3.5" Floppy Drive, or as many as 300 or more, as in the case of the C Ltd. 800 Megabyte Optical Disk. Under AmigaDos. Each BLOCK is made up of 512 Bytes (256 Bytes in the case of the IOmega Bernoulli Box) of information.

INTERLEAVE refers specifically to the order that the BLOCKs of data are placed on the disk. The diagram at the right displays one TRACK with 11 BLOCKS from an Amiga 3.5" Floppy Disk. The TRACK as shown has an INTERLEAVE Factor of 1:1 (1 to 1), that means that the BLOCKs of data are put on the disk in a sequential order as shown in the graphic. If we wanted to get a program from this disk that was eight BLOCKs long and started at BLOCK # 2, the drive would read BLOCK # 2, then BLOCK # 3, then BLOCK # 4, etc. until all of the program had been read.

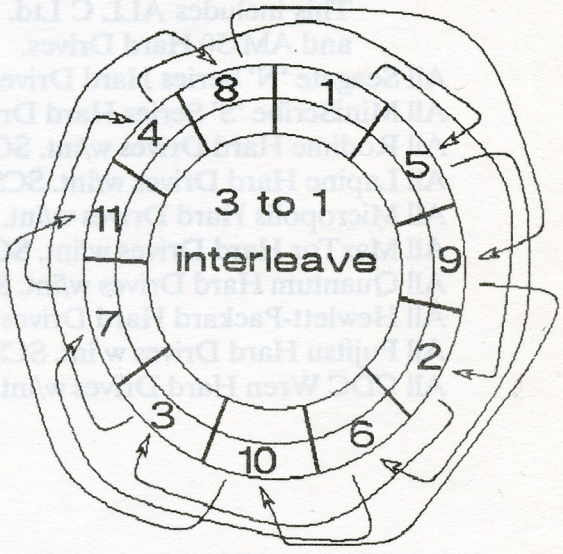


Though looking well organized, there is a potential problem with putting data on the disk in this way. The Amiga usually can't recieve the data from a Hard Drive as fast as the Hard Drive can deliver it! In practice, that would mean that the drive would have to (as in the above example) get the data from BLOCK # 2, send the data to the Amiga computer, wait for the Amiga to tell the drive that it was ready for the next BLOCK, then the drive would have to read that BLOCK. This would **ALL** have to happen before the disk rotated from the ending point of BLOCK # 2 and the starting point of BLOCK # 3. Most of the time it just won't be ready fast enough and the computer will then have to wait for the drive platter (disk) to make another **entire revolution** to get back to the starting point for BLOCK # 3. Waiting that long on every BLOCK read would mean that the drive would have to make one complte revolution for each BLOCK to be read and would make a noticeable difference in the speed of operation of the drive.

To solve this problem drive INTERLEAVE was developed. If we put BLOCK # 1 at position 1, then skipped one position and put BLOCK # 2 in position 3 (this would be an INTERLEAVE Factor of 2:1), we would now have gained the time it takes for the disk to rotate from the end of position 1 to the beginning of position 3 to allow the Amiga to move the data and get ready for the next read. This would allow the entire TRACK to be read in just TWO revolutions, instead of 11. At the left is a diagram of the same CYLINDER as before, but changed to reflect the skipping of one position (2:1 Interleave) between each BLOCK of data.



If skipping one position (INTERLEAVE Factor of 2:1) still doesn't give us enough time, we could skip two positions (INTERLEAVE Factor of 3:1) and double the time between ending the read and having to be ready to read in the next BLOCK. Shown to the right is the same CYLINDER as before, but with an INTERLEAVE Factor of 3:1



We could get even more time by using INTERLEAVE Factors of 4:1, 5:1, 6:1 or more. In fact you can use INTERLEAVE Factors of up to one less than the total number of BLOCKs per CYLINDER allowed by your drive. The objective of using INTERLEAVE is to speed up the operation of the disk drive by eliminating as much wait-time as possible between the reading of BLOCKs of data.

Now that you know what the INTERLEAVE Factor is, how do you determine what INTERLEAVE Factor will give you the best performance. Well, due to large variations in drive rotational speed, head movement, and electronics performance, the only method that works is TRIAL AND ERROR. And since setting the INTERLEAVE Factor can only be done during the low-level formatting of the drive, it is a truly time consuming chore to try many different INTERLEAVE Factors.

If you really want to optimize your Hard Drive's interleave, bring your lunch! (To help speed up the process, we would recommend that you only format a small section of the drive, for example the first 20 cylinders, for testing purposes.)

Suggested Interleaves:	*SCSI*	*KRONUS*
All ST-506 Type Hard Drives with Adaptec ACB-4000 Controllers: This includes most C Ltd. AM-20, AM-22, AM-24 and AM-80 Hard Drives, EXCEPT those units sold as Hi-Speed Hard Drives.	2:1	1:1
All ST-506 Type Hard Drives with Adaptec ACB-4070 Controllers: This includes most C Ltd. AM-30, AM-33, AM 44 and AM-50 Hard Drives, EXCEPT those units sold as Hi-Speed Hard Drives.	5:1	1:1
All ST-506 Type Hard Drives with Omti 3520 Controllers: This includes ALL C Ltd. Hi-Speed AM-20, AM-22, AM-24 and AM-80 Hard Drives.	1:1	1:1
All ST-506 Type Hard Drives with Omti 3527 Controllers: This includes ALL C Ltd. Hi-Speed AM-30, AM-33, AM-44 and AM-50 Hard Drives.	1:1	1:1
All Seagate 'N' Series Hard Drives w/int. SCSI:	2:1	1:1
All MiniScribe 'S' Series Hard Drives w/int. SCSI:	2:1	1:1
All Rodime Hard Drives w/int. SCSI:	2:1	1:1
All Lapine Hard Drives w/int. SCSI:	2:1	1:1
All Micropolis Hard Drives w/int. SCSI:	1:1	1:1
All MaxTor Hard Drives w/int. SCSI:	1:1	1:1
All Quantum Hard Drives w/int. SCSI:	1:1	1:1
All Hewlett-Packard Hard Drives w/int. SCSI:	1:1	1:1
All Fujitsu Hard Drives w/int. SCSI:	1:1	1:1
All CDC Wren Hard Drives w/int. SCSI:	1:1	1:1

SCSInet™

Of particular interest to many is the multi-user networking capability presented by the SCSI system. C Ltd. is now developing the required software and drivers to make possible true multi-tasking/multi-user systems based on your Amiga computer. The Amiga Multi-User system software should be a reality soon. Together with several of the major Amiga software suppliers C Ltd. is working to provide a standardized environment for business applications of the Amiga.

If you are interested in an inexpensive network system that you can start using today, read the rest of this section carefully. At the present time the C Ltd. SCSInet is limited to performing file server functions. This means that you can connect several Amiga Computers to one or more Hard Drives, but with some limitations, most of which can be easily gotten around. The C Ltd. SCSInet file server system will allow complete read/write access by any computer on the system to any file on any drive, but NO FILE PROTECTION has been implemented. Due to the way AmigaDOS works, writing to the hard drive by multiple computers is somewhat complicated and not always safe, but there are several ways around this problem.

The biggest problem stems from the fact that AmigaDOS loads the bitmap (where information is stored) from all devices when they are mounted and then stores the bitmap in the Amiga's RAM memory, updating the information in its own memory as the computer makes additions and/or deletions to files on the device, then writing the updated bitmap information to the device. The problem is that the Amiga relies strictly on the bitmap information in its own memory to read and write data to the device and never re-confirms the bitmap information by re-loading it into memory. In other words, AmigaDOS is totally un-aware of any changes that might be made to the bitmap by any device or computer other that the changes it makes itself.

For example, let us define a system with Computer(A) and Computer(B) connected to Drive(C). We turn on Drive(C) and Computer(A), and start working with Computer(A). First we load an editor, then a file to edit. When we are done with the file we save it (and the Amiga stores an updated bitmap to Drive(C)) and load a new file. (So far we're O.K.) Now, while we are working on the second file, someone comes along and turns on Computer(B) and mounts Drive(C), the Amiga loads the current bitmap from Drive(C) into its memory, and the second user now loads a spread sheet program and a spread sheet data file from Drive(C). (So far we're still O.K.) After finishing our editing session at Computer(A), we store the file on Drive(C) and Computer(A) again writes an up-dated bitmap to Drive(C). Now, we have a problem because the bitmap information in stored Computer(B)'s memory doesn't match the actual bitmap of Drive(C) which was just changed by Computer(A) when we stored our work. As long as we read and write data from Computer(A), we're O.K. We're even O.K. if Computer(B) tries to read data from Drive(C) in which case, if the bitmap in Computer(B)'s memory doesn't match the actual bitmap, Computer(B) either assumes someone changed disks without telling it and goes ahead and re-loads the new bitmap information, or thinks that the disk has been corrupted and issues an error message. BUT.... if the operator at Computer(B) attempts to store data to Drive(C), Computer(B) will, without any checking or warning, use the bitmap in its memory

(which doesn't match the actual bitmap of drive) to place the data on Drive(C), and this will result in a total invalidation of Drive(C) rendering it unusable.

There are three ways around this problem. First, every partition of a hard drive is treated as a seperate device and the bitmap information for each is maintained seperately, so you could partition your hard drive into enough partitons so that each computer connected to the system had its own partition. You would then restrict each operator to write data only to his/her own partition. Every computer/operator could have full read access to any partition, but for whatever reason, if desired, you could restrict access to any partition from a specific computer by simply not including the partiton information in the MountList file on the boot disk in that computer. In this case if an operator wanted to work on a file in another operator's partition, the operator would copy the file from the desired partion into his/her own, do the work and then notify the operator who originated the file who would then, working from the computer authorized to write to the partition where the file was originally stored, copy the file back into its original partition. While this is not a perfect solution, it is simple, direct and workable for many installations.

The second solution is even simpler than the first, but requires much more care on the part of the operators and would be totally unsuitable for installations where data flow is not carefully controlled. If, as stated before, the problem results from using an erronious bitmap to write data to a device, why not just update the bitmap every time before you write to a device? The AmigaDOS "DISKCHANGE" command will force the computer to re-load the current bitmap from any device and therefore could be used before every write access to insure that the most current bitmap is being used. In many situations this will work fine, but it is possible that another computer can write to the device, thereby altering the bitmap, in the few seconds between the time that the DiskChange command forces the bitmap update and the time that the file is actually written to the device. So, if you are using the DiskChange command, you should use extreme caution to insure that the bitmap is not updated after issuing the DiskChange command, but before the data is written.

The final solution is involved, but will produce wonderful results in some situations. If your system requires constant updating of a "master" file, as in data base mainainence, from several computers look at this as a possibility. Partition your hard drive into enough partitons so that each computer connected to the system had its own partition. Every computer/operator could have full read access to the "master" file, but modify the program being used so that it will write the update information as a "special" packet of data into the partition authorized for that computer's write access. Finally, write a separate program that runs as a background task in one of the computers that continually scans all of the partitions looking for the "special" data packets and uses them to update the "master" file.

NOTE: Due to the lack of file locking capabilities within the AmigaDOS operating system and the general nature of the SCSInet system at its present stage of development, C Ltd. will disclaim any liability resulting from its use, and urge users dealing with critically important data to aviod using SCSInet unless they back-up data regurlarly and have the ability and are fully prepared to recreate such data if it is lost.

C Ltd. Hard Drive Specifications

Power Requirements.

C Ltd.'s SCSI Host/Controllers for the Amiga 1000 and 2000 require approx. 450ma. of power, KRONOS Host/Controllers for the 1000 and 2000 require approx. 650ma. of power, either which is drawn from the Amiga computer's expansion bus. The SCSI for the Amiga 500 requires approx. 250 ma. of power, the KRONUS unit requires approx. 400ma. of power, either of which is drawn from the Amiga 500's expansion bus. The Amiga 1000 has 1000ma. available at that port, the 500 has approx. 100 ma. available and the Amiga 2000 has 2500 ma. available. If you have an Amiga 1000 or an Amiga 500 and you have other devices on the expansion port that also draw power from it, you should be sure that you are not exceeding the Amiga's limits.

SCSI Host/Controller Software Interface.

The C Ltd. SCSI Host/Controller complies with the standards described for Small Computer System Interface (SCSI) Devices. (American National Standards Institute - ANSI SCSI Standards.)

Data Transfer Rate.

The burst data transfer rate (read/write) for most imbedded SCSI Hard Drive Systems and ST-506 devices with SCSI ST-506 device controllers is approx. 5 Megabits/sec. or 625 Kbytes/sec.

The burst data transfer rate for imbedded SCSI Hard Drive Systems and ST-506 devices with SCSI ST-506 device controllers using RLL data transfer technology is approx. 7.5 Megabits/sec. or 937.5 Kbytes/sec.

The burst data transfer rate for imbedded SCSI Hard Drive Systems and ESDI devices with SCSI/ESDI device controllers using ESDI data transfer technology is approx. 15 Megabits/sec. or 1.875 Megabytes/sec.

When used with a standard Amiga A-500, A-1000 or A-2000, the C Ltd. SCSI series controllers can sustain burst transfers of 600 Kbytes/sec. and sustained transfers of 350 Kbytes/sec. KRONUS series controllers are capable of bursts of 1.2 Megabytes/sec. and sustained transfers of 600 Kbytes/sec. When used with Amiga computers outfitted with **68020/68030** processors the SCSI series controllers can sustain burst transfers of **1.2 Megabytes/sec.** and sustained transfers of **650 Kbytes/sec.** KRONUS series controllers are capable of bursts of **2.0 Megabytes/sec.** and sustained transfers of **1.0 Megabytes/sec.**

NOTICE:

Special applications using the *SCSI-Link Library* may take advantage of the C Ltd. SCSI series Controller's ability to transfer data at sustained speeds of 1.2 Megabytes/sec. or the KRONUS series at 2.0 Megabytes/sec.

NOTE: Normal multi-tasking operations under Amiga DOS 1.3 may degrade data transfer of all Hard Disk Drive systems, but severe speed degradation, similar to that of DMA type SCSI controllers when used in conjunction with hi-res overscan screens, is not experienced under any condition due to the pseudo-DMA/buffered-programmed-IO transfer design employed in both the SCSI and KRONUS controllers.

Method Of Operation:

Both the SCSI series and KRONUS series controllers use the NCR/AMD 5380 SCSI Interface IC operating in its DMA mode to transfer data into a buffer on the C Ltd. SCSI controller. Data is then transferred into the Amiga using the standard interrupt driven programmed I/O method. This system offers the best attributes of both methods. Using DMA transfer to achieve the fastest possible data aquisition from SCSI devices, while using processor (rather than DMA) transfer from the SCSI controller into the Amiga eliminates possible system speed degradation caused by the Amiga's custom processors (Agnes, Paula and Denise) as they control DMA bus activity.

Using the SCSI Host/Controller with Other SCSI Devices.

In the near future, if there is enough interest, C Ltd. will be releasing additional devices that will connect to the SCSI Bus, and more special software to address other types of SCSI devices. Most of the newest technology devices are being released in the SCSI format, and many of these will be available **only** as SCSI devices. While SCSI is a standard, each type of device connected to the SCSI system bus may require some different software to address the special features of these devices. To augment C Ltd.'s ability to keep up with these required software drivers, we have releases the SCSI-Link Library as a programmers tool. By using the Library, any competent programmer will be able to write software drivers able to support any possible SCSI device that may be released.

Currently available, in development or under consideration are the following SCSI devices:

AVAILABLE NOW:

> an SCSI 10 Meg removable media Hard Drive system
> an SCSI 44 Meg removable media Hard Drive system
> an SCSI 60 Meg Streaming Tape Back-up System.
> an SCSI 800 Meg Optical Disk Mass Storage System.
> an SCSI High Speed 300 DPI Laser Printer w/PostScript.
> an SCSI 300 DPI Optical Scanner with Optical Character Recognition.

Under Development:

> an SCSI Multi-User (Amiga) Networking System.
> an SCSI Higher Speed Post Script Laser Printer.
> an SCSI CD ROM Mass Storage System.
> more removable media devices.a

Under Consideration:

> an SCSI Laser Disk Interface.
> an SCSI Non-Volatile RAM Disk. (Super Fast 2MB/sec.)
> an SCSI Amiga to I.B.M. SCSI Link.
> an SCSI Real-World Data Acquisition System.
> an SCSI to Multiple Parallel Printer Port Interface.
> an SCSI to Multiple RS-232 Port Interface.

If you have a specific need for any of these hardware products and/or software drivers, contact our Customer Service Department. We can and will do custom drivers for special applications.

Potential Problems and/or Unexpected Events.

Built-in Hard Drive Validation.

All C Ltd. Hard Drives have a built-in mechanism that will re-establish data on the disk after some forms of crashes, meeting the Guru while writing to the Hard Disk for example. The C Ltd. Hard Drive has built in diagnostics and self-tests that will correct many problems while they are still minor enough to be fixed. So, if at some point in time your drive looks as if it has locked-up or gets stuck, don't be impatient and re-boot, or turn the drive off. If the LED (the little light on the front of the drive) is flashing feverishly, or blinks on for a second or two then goes off for a while, then blinks again, wait a couple of minutes and your drive may magically fix itself. If it doesn't, before you go off and re-format the drive and destroy all of your data, try this sequence:

1. If possible save any files in RAM to floppy disk.
2. Re-boot using Amiga-Amiga-Ctrl.
3. Wait at least thirty minutes. (Much longer on larger drives.)
4. If the system doesn't come up then, turn off everything.
5. Restart the system from KickStart.
6. After inserting your Boot disk, wait at least thirty minutes.
7. If the system doesn't come up then, re-boot again using Amiga-Amiga-Ctrl.
8. Wait at least thirty minutes.
9. If the system doesn't come up then, re-boot using Amiga-Amiga-Ctrl and as soon as the LED on the Hard Drive blinks the first time, quickly re-boot again.
10. Wait at least thirty minutes.
11. If the system doesn't come up then, repeat the complete sequence again.

If it isn't up by now...Give Up! Re-format your drive.

If you have more that one device to plug into the Amiga expansion port.

First, the C Ltd. SCSI Host/Controller should work with any other device on the expansion port, unless the manufacturer of the device specifically has prevented such operations by failing to follow the published Amiga specifications for devices that are to be attached to the Amiga Expansion Bus. The C Ltd. SCSI Host/Controller is a true Auto-Configuring device and as such conforms completely to Commodore-Amiga specifications for expansion devices. The C Ltd. SCSI Host/Controller is not sensitive to position, if several devices are plugged into the expansion bus, the SCSI Host/Controller may be the first, second, third or wherever you find it convenient.

Also you should be aware that SOME (10-20%) of the Amigas in existence experience problems when a second device is added to the expansion port. Commodore acknowledges this problem exists, but offers no solution. If Your computer works fine with the C Ltd. SCSI Host/Controller connected all by itself, and it works fine with another device (like a RAM card) connected all by itself, but won't work with both of them connected at the same time, you should call our Customer Service Department at (316) 267-0202 for instructions. We can offer several solutions to this particular problem, including high speed IC chips to replace your original problem ICs with parts that work, we have these chips and will make them available to our customers.

No Icon shown for the Hard Drive after Loading WorkBench.

The Amiga's Workbench will not display an icon for any device or partition added to your system (eventhough it has been properly mounted) until after the first time that the device or partition is actually accessed. Any command that addresses the device or partition may be used to trigger the icon, for example:

1> DiskChange DH0: <RETURN>

1> CD DH0: <RETURN>

1> Dir DH0: <RETURN>

1> Run DH0:Demo <RETURN>

1> Assign c: DH0:c <RETURN>

Any of the above commands (or any similar commands) will be sufficient to cause the Amiga WorkBench to display the icon for your device or partition on the WorkBench screen.

Calling Customer Service

If you are confused, lost or just plain need help because something doesn't work the way you want it to, your first call should be to the dealer who sold you your system. Your dealer is your first line of support for several very good reasons. First, he/she sold you the system and therefore is probably much more familiar with the various components and software that you own. Second, your dealer is usually a local (toll free) call. And last, your dealer knows you and understands your needs better that we could. C Ltd. products are sold exclusively through qualified Amiga dealers for exactly the above reasons.

We do, however, recognize that there are questions that your dealer will not be able to answer and/or problems that he/she can do nothing about. In these cases do not hesitate to call our Customer Service line for help. Making sure that our products perform properly is our Number One priority. It is an expensive undertaking to offer outstanding customer support, but that is our goal. C Ltd. wants to offer every customer the individual help he/she needs to get their specific problems solved. We would like to ask your help in reaching our goal, because in reality helping us is helping yourself in this case. Here are a few simple things that you can do to help:

First, before you call Customer Service, be sure that you have read this instruction manual thoroughly and have followed the instructions properly. Historically we have found that a large percentage of calls to our Customer Service line would not have been necessary if the customer had simply read the manual. In our opinion, it is unfair to take our Customer Service agent's time away from those customers who have real need of the help to provide information that is in the manual. On the other side of the coin, we are not saying that our manual is perfect and contains all the answers, or that the information in it is so well written that it should be "clear as a bell" to everyone. If you are totally confused, please, do call us. All we're asking is that you give the information in the manual a fair shot first.

Second, be prepared when you call. Put down on paper all the questions, problems and as many relevant facts as you can think of. Have a pencil and paper ready. If the problem involves another manufacturer's product, have their instructions and documentation ready. And last but by no means least, ***IF AT ALL POSSIBLE,*** be at your computer when you call. Trouble shooting problems is tough over the phone, but following these guidelines will substantially increase our odds of solving the problem quickly and efficiently.

Third, we'll be patient with you, please be patient with us. Our policy is that each caller to Customer Service will receive all of the time he or she needs to explain any problems and to receive and understand the solutions. We will even take the time to walk you through the solution, right on your own computer, if it will help. Because of the time spent with each caller, you might find our phone lines busy from time to time. As much as you detest hearing that busy signal, remember that we are taking the time to help some one, and when it is your turn, you will appreciate all the time we'll spend with you as well.

NOTE: We also monitor Compuserve and will provide answers to questions about our products posted in the Amiga Forum in our section. To contact us on Compuserve, go to the AMIGA FORUM and post your messages in the C Ltd. area. We also maintain our own 24 hour, 300/1200/2400 baud BBS at (316) 267-1222.

Other Utilities and Other Special Programs

In addition to the programs that are necessary to the operation of SCSIdos_3.0 we have included several programs for educational and diagnostic uses. The following is a list of the additional Utility programs available on the SCSIdos_3.0 Disk. If any of the programs mentioned herein is not on your disk, they are all available for download from the C Ltd. BBS or will be sent to you free if you send us a list of the programs that you want and $3.00 to cover the disk cost and the postage and handling.

Inquire

Inquire is a WorkBench Icon operated program that when selected (double click on the Inquire Icon) will poll all of the devices on the SCSI bus and will return the SCSI address and description of any SCSI devices found.

Inquir

Inquir is a CLI operated version of the above program.

FORMAT: INQUIR <RETURN>

Check

Check is a WorkBench Icon operated program designed as a diagnostic tool for testing Adaptec SCSI/ST-506 device controllers. When selected (double click on the Check Icon) the program will ask the user for the SCSI address of the SCSI device to be checked and will then test the C Ltd. SCSI Host/Controller (returning its status and ID number) and test the Adaptec by writing to and reading from the RAM buffer on the Adaptec, and finally requesting the Adaptec's device status. If the Adaptec unit is found and the RAM test is O.K. the Adaptec controller found message will be displayed, but if the drive attached to it is defective or not formatted, an error message will also be returned.

SCSIcheck

SCSIcheck is the CLI version of the above program.

FORMAT: SCSIcheck (SCSIaddress) <RETURN>

Size

Size is a WorkBench Icon operated program that when selected (double click on the Check Icon) will ask the user for the SCSI address of the SCSI device to be checked and will then return the capacity of the device (in sectors) if the device exists and is capable of providing the required information.

CAP

CAP is the CLI version of the above program.

FORMAT: CAP (SCSIaddress) <RETURN>

Clear

Clear is a WorkBench Icon operated program that when selected (double click on the Check Icon) will ask the user for the SCSI address of the SCSI device to be cleared and will then send the SCSI RESET command to that device. The program will return the device status if the device exists and is capable of providing the required information.

ClearSCSI

ClearSCSI is the CLI version of the above program.

FORMAT: ClearSCSI (SCSIaddress) <RETURN>

Errors

Errors is a WorkBench Icon operated program that when selected (double click on the Check Icon) will ask the user for the SCSI address of the SCSI device to be tested and will then perform a non-destructive error test of the device by reading (data and checksums) of every sector on the device. If any errors are encountered they are reported in both their sector number format and their track/head/byte-offset formats. Additionally, 32 retries are made to read any defective sector, if the sector can be read the actual number of retries required is reported, otherwise the maximum of 32 retries is reported.

When used as a diagnostic this tool can provide valuable data. If any sectors require more than 3-6 retries, a total Low-Level reformatting of the device is indicated. If on several run throughs of the Errors program random errors (with a low number of retries) are reported that are not consistant from run to run, the test suggests noise is being introduced into the system from a noisy power supply (common on A-2000s) or that there is a cold solder joint, dirty contact, intermittant cable connection or other erratic hardware problem.

ReadErr

ReadErr is the CLI version of the above program.

FORMAT: ReadErr (SCSIaddress) [lowSector] [hiSector] <RETURN>

Formatter

Formatter is a WorkBench Icon operated program that when selected (double click on the Check Icon) will ask the user for the SCSI address of the SCSI device to be formatted and the Interleave to be used. The program will then send the SCSI FORMAT command to that device. The program will return the device status if the device exists and is capable of providing the required information. This format command will format ANY SCSI device capable of being formatted, but will not map out any errors! In general if the normal FEdX formatter will not format a drive, try this one, but be sure to run the Errors program on it to see if there are any errors that could cause problems.

GenFormat

GenFormat is the CLI version of the above program.

FORMAT: GenFormat (SCSIaddress) (Interleave) <RETURN>

Additional Utility programs of this type are available on the C Ltd. BBS. It is our intention to continually update and improve SCSIdos and these programs are part of that process. If you have any suggestions for improvements of these programs, or for other programs of a similar nature, please write us or post your ideas on our BBS.

The SCSI-Link Library.

To the best of our knowledge, C Ltd.'s SCSIdos_3.0 is the only SCSI software that can be used by the general public to access SCSI devices. We are the only company that actually publishes and supports a standard format AmigaDOS type Library that allows complete unlimited access to the SCSI bus and any devices on it. Please don't abuse this software by writing in your own non-standard calls, trashing your hard drive and then blaming it on us! The SCSI-Link Library gives you access to SCSI devices at their lowest level, so you can wreak havoc if you are not careful, but you can work wonders if you do it right. You could write the world's fastest database by setting AmigaDOS aside and using the actual SCSI sectors as records and indexing them directly, or you can do real-time animation directly off the hard drive by by-padding AmigaDos and loading uncompressed ILBM images directly into a double buffered screen structure. The following is a brief description of the SCSI-Link Library and some sample programs using it, additional information is available on our BBS.

You should have (or get from the C Ltd. BBS) the following files:

```
SCSI-Link.h --------- Main Library Header File
SCSI-Link f.h ------- SCSI Link Function Definitions

SCSI-Link.lib ------- Object Module To Link With Lattice C
SCSI-Link.library --- Amiga Format Library (Put Into LIBS: Directory)

Inquire.c -------- Test program for the library.
Compile ---------- Used to compile test program (execute Compile Inquire).
Link ------------- Used to link test program (execute Link).
LinkFile --------- Link description file.

ShowSize.c ------- Sample function for use with library.
```

Sorry for the current lack of documentation. We are hard at work attempting to finish our KRONOS controller. We plan on providing better docs in the future.

I currently only have support for the Lattice complier, will include support for Manx later.

A few notes:

This library if for version 3.x ONLY.

The DevInstall program must have already been run before opening the library.

The HandlerNumber argument in the open request structure refers to the devsetup entry handler number (Current standard is 100).

At least one unit (HardDisk) MUST be open that uses the specified handler before attempting to open ANY unit thorugh the library.

```
/* ------------------------------------------------------------------ */
*                                                                     *
* SCSI-Link.i -- SCSI Link Library Public Structures                  *
*                                                                     *
*                      Wayne E.  Miller                               *
*                      October, 1988                                  *
*                                                                     *
* Copyright (C) 1988, Wayne E.  Miller - All rights reserved.         *
*                                                                     *
* ------------------------------------------------------------------ */

/* --- Library Name Macro ------------------------------------------- */

#define LB NAME    "SCSI-Link.library"
/* --- Open Unit Parameter Structure -------------------------------- */

extern struct OpenDevs {            /* Open Unit Parameter Structure    */
UWORD    ou HandlerNumber;        /* Handler Number                  */
UBYTE    ou Address;              /* Unit Address                    */
UBYTE    ou Flags;                /* Open Flags                      */
};


/* --- SCSI Command I/O Request Structure --------------------------- */

extern struct SCSI IO {             /* SCSI I/O Request Structure       */
UWORD    sio Command;             /* Command Sequence Type           */
UBYTE    *sio CmdBuffer;          /* Pointer To Command Buffer       */
ULONG    sio CmdLength;           /* Length Of Command               */
UBYTE    *sio DataBuffer;         /* Pointer To Data Buffer          */
ULONG    sio DataLength;          /* Transfer Length (Bytes)         */
UBYTE    sio Status;              /* Status Byte                     */
UBYTE    sio Message;             /* Message Byte                    */
};

/* --- Disable Interrupts During I/O -------------- */

#define    SLB DISABLE IRQ     7
#define    SLF DISABLE IRQ     (1<<7)

/* --- Request Exclusive Access ------------------- */

#define    SLB EXCLUSIVE       5
#define    SLF EXCLUSIVE       (1<<5)

/* --- Allow Unit To Be Defined (DevSetup) ---------- */
```

```
#define     SLB DEFINED             2
#define     SLF DEFINED             (1<<2)

/* --- Allow Unit To Be Opened (CLtd.device) --------- */

#define     SLB OPENED              1
#define     SLF OPENED              (1<<1)

/* --- Allow Multiple Direct Accesses (SCSI-Link) ---- */


#define     SLB MULTI               0
#define     SLF MULTI               (1<<0)

/* --- Command Sequence Definitions ------ */

#define     SCSI SendCommand    4     /* Send Command To SCSI Device          */
#define     SCSI GetInfo        8     /* Get Information From SCSI Device     */
#define     SCSI PutInfo        12    /* Send Information To SCSI Device      */
#define     SCSI ReadData       16    /* Read Data, Single Byte Multiples     */
#define     SCSI WriteData      20    /* Write Data, Single Byte Multiples    */
#define     SCSI ReadBlocks     24    /* Read Data, 256 Byte Multiples        */
#define     SCSI WriteBlocks    28    /* Write Date, 256 Byte Multiples       */
```

```
/* ------------------------------------------------------------------ *
*                                                                     *
* SCSI-Link f.h -- SCSI Link Libary Function definitions              *
*                                                                     *
*                      Wayne E.  Miller                               *
*                      October, 1988                                  *
*                                                                     *
* Copyright (C) 1988, Wayne E.  Miller - All rights reserved.         *
*                                                                     *
* ------------------------------------------------------------------ */

extern char *SL OpenUnit();             /* Open Unit                   */

extern int  SL DoSCSI();                /* Do SCSI Command             */

extern void SL CloseUnit();             /* Close Unit                  */

/* ------------------------------------------------------------------ *
* ShowSize.c -- Show Device Capacity                                  *
*                                                                     *
*                      Wayne E.  Miller                               *
*                      October, 1988                                  *
*                                                                     *
* Copyright (C) 1988, Wayne E.  Miller - All rights reserved.         *
* ------------------------------------------------------------------ */

/* --- System Includes -------------- */

#include "exec/types.h"
#include "exec/nodes.h"
#include "exec/lists.h"

#include "stdio.h"

/* --- Program Includes ------------- */

#include "Current:SCSI-Link.h"
#include "Current:SCSI-Link f.h"

#include "/Formatter.h"

/* --- Declare External Functions ---- */

extern UBYTE GetSense();                /* Get Device Sense Data       */
extern void ProgEnd();                  /* Program Ending Routines     */

/* --- Data Structures -------------- */
```

```
extern struct DriveCapacity {
ULONG    BlockAddress;
ULONG    BlockSize;
};

/* --- Function Entry Point ---------------------------------------------- */

void ShowSize()
{
static UBYTE  CmdBuffer[] = { 37, 0, 0, 0, 0, 0, 0, 0, 0, 0 };

UBYTE  Result;

struct DriveCapacity dc;

sio.sio Command     = SCSI GetInfo;
sio.sio CmdBuffer   = CmdBuffer;
sio.sio CmdLength   = 10;
sio.sio DataBuffer  = (UBYTE *) &dc;
sio.sio DataLength  = sizeof(struct DriveCapacity);

Result = DoSCSI(sc, &sio);

if (Result != 0)
ProgEnd("Unit Select Failure\n");

if (sio.sio Status != 0) {
Result = GetSense();
ShowError(Result);
ProgEnd("Device Error");
}

fprintf(stdout, "\n Block Size      - %dl Bytes\n",
dc.BlockSize);

fprintf(stdout, "\n Drive Capacity - %ld Blocks, (%ld Bytes)\n",
dc.BlockAddress, (dc.BlockAddress * dc.BlockSize));
}
```

```
/* ------------------------------------------------------------ *
* Inquire.c -- SCSI Link Library Test Program                   *
*                                                               *
*                     Wayne E.  Miller                          *
*                     October, 1988                             *
*                                                               *
* Copyright (C) 1988, Wayne E.  Miller - All rights reserved.   *
* ------------------------------------------------------------ */

/* --- Version & ID Stuff ------------------------------------ */

#define VERSION    3
#define REVISION   0

/* --- System Includes --------------------------------------- */

#include "exec/types.h"
#include "exec/nodes.h"
#include "exec/lists.h"
#include "exec/memory.h"
#include "exec/libraries.h"

#include "stdio.h"
#include "string.h"
#include "dos.h"

/* --- Program Includes -------------------------------------- */

#include "Current:SCSI-Link.h"
#include "Current:SCSI-Link f.h"

/* --- Declare Internal Functions ---------------------------- */

int ProcBreak();                    /* Break Trap Processor      */
int ProcBreak2();                   /* Alternate Break Trap      */

void ProgEnd();                     /* Program Cleanup Routines  */

/* --- System Variables -------------------------------------- */

struct library *ExecBase = 0;           /* Exec Base Pointer     */
struct library *DosBase = 0;            /* Dos Library Pointer   */
struct library *SCSILinkBase = 0;       /* Library Base Pointer  */

/* --- General Purpose Variables ----------------------------- */

ULONG   UnitNumber = 0;                 /* Unit Number           */
```

```
struct  OpenDevs *ou = 0;              /* Open Unit Parameters     */
struct  SCSI IO *sio = 0;              /* SCSI I/O Request Struct  */

struct  SCSI Control *sc = 0;          /* Library Control Pointer  */

/* --- Program Entry Point ------------------------------------- */

void main()
{
char Buffer[100];

ULONG   Result, Index;

static UBYTE Command[] = { 18, 0, 0, 0, 36, 0 };

/* --- Print Program Identification ---------------------------- */

fprintf(stdout, "SCSI Device Inquire Command - Version %u.%u \n\n",
VERSION, REVISION);

/* --- Set Up Break Trap --------------------------------------- */

if (onbreak(&ProcBreak) != 0)
ProgEnd("Unable To Set Break Trap");

/* --- Get / Allocate System Resources ------------------------- */

if ((SCSILinkBase = (struct library *) OpenLibrary(SL NAME, 0)) == 0)
ProgEnd("Unable To Open SCSI-Link Library");

if ((ou = (struct OpenDevs *) AllocMem (sizeof(struct OpenDevs),
MEMF PUBLIC|MEMF CLEAR)) == 0)
ProgEnd("Memory Allocation Failure");

if ((sio = (struct SCSI IO *) AllocMem (sizeof(struct SCSI IO),
MEMF PUBLIC|MEMF CLEAR)) == 0)
ProgEnd("Memory Allocation Failure");

/* --- Main Loop ----------------------------------------------- */

for (UnitNumber = 0; UnitNumber < 8; UnitNumber += 1) {

ou->ou HandlerNumber = 100;
ou->ou Address = UnitNumber;
ou->ou Flags = SLF DEFINED|SLF OPENED|SLF MULTI;

fprintf(stdout, "SCSI Address - %d - ", UnitNumber);
```

```
if ((sc = OpenUnit(ou)) == 0) {
fprintf(stdout, "[ Unable to Open Unit ]\n");
continue;
}

sio->sio_Command = SCSI_GetInfo;
sio->sio_CmdBuffer = Command;
sio->sio_DataBuffer = Buffer;
sio->sio_DataLength = 300;

Result = DoSCSI(sc, sio);
CloseUnit(sc);
sc = 0;

while(TRUE) {

if (Result != 0) {
fprintf(stdout, "[ Unit Select Failure ]\n");
Delay(5 * 50);
break;
}

if (sio->sio_Status != 0) {
fprintf(stdout, "[ Unit Status Error ]\n");
break;
}

for(Index = 0; Index < sio->sio_DataLength; Index++) {

if ((Buffer[Index] < 32) || (Buffer[Index] > 126))
Buffer[Index] = ' ';
}

Buffer[Index] = '\0';
fprintf(stdout, ">%s<\n", Buffer);

break;
}
}
ProgEnd(0);                    /* Normal Program Exit              */
}

int ProcBreak()
{
if (onbreak(&ProcBreak2) != 0)
ProgEnd("Unable To Set Alternate Break Trap");

ProgEnd("User Request");
```

```
return(0);
}

int ProcBreak2()
{
return(0);
}

void ProgEnd(ErrorText)
char *ErrorText;
{
/* Display Error Message If Needed */

if (ErrorText != 0)
fprintf(stdout, "\nProgram Aborted - %s\n", ErrorText);
else
fprintf(stdout, "\nProgram Finished\n");

if (sc != 0)
CloseUnit(sc);

if (sio != 0)
FreeMem(sio, sizeof(struct SCSI IO));

if (ou != 0)
FreeMem(ou, sizeof(struct OpenDevs));

if (SCSILinkBase != 0)
CloseLibrary(SCSILinkBase);

if (ErrorText == 0)
exit(0);
else
exit(100);
}
```

```
***************
Sample Compile
***************


.k file/a
IF EXISTS <file>.c
ECHO ""
ECHO " # # # # #  Processing File <file>.c  # # # # # "
ECHO ""
lc:c/lc1 -cst -oram: <file>
lc:c/lc2 -v -oram: ram:<file>
Copy ram:<file>.o To <file>.o
Delete ram:<file>.o
ELSE
ECHO ""
ECHO " # # # Error # # #  File <file>.c Does Not Exist  # # # "
ECHO ""
ENDIF


************
Sample Link
************


lc:c/blink WITH LinkFile

Copy ram:Inquire Inquire


*****************
Sample Link File
*****************

FROM        lc:lib/c.o,
Inquire.o

TO          ram:Inquire

LIBRARY     lc:lib/lc.lib
lc:lib/amiga.lib
Current:SCSI-Link.lib

MAP         nil:
```

Index

25 Conductor Cable TRM-35
256 Byte Blocks Page 56
3.5 inch Form Factor Hard Drives Page 14
Page 20
5.25 inch Form Factor Hard Drives Page 14
Page 20
50 Conductor Cable TRM-35
512 Byte Blocks Page 56
A-1000 Bus Pass-Thru Page 9
TRM-37
Adding Devices To The SCSI Bus TRM-34
Adding Expansion Devices, A-1000 Page 9
Additional SCSI Devices TRM-34
Address, C Ltd. Page 1
AmigaDOS Format Page 62
AmigaDOS TRM-4
ANSI SCSI TRM-34
Apple/MacIntosh SCSI TRM-34
APTB Instsallation TRM-37
APTB Pass-Thru Board Page 9
TRM-37
Arbitration TRM-7
TRM-8
Auto-Start TRM-11
TRM-14
AutoRetries TRM-25
Back-Up, Hard Drive TRM-38
Backing Up SCSIdos 3.0 Page 5
BBS, C Ltd. Page 1
Bernoulli Box Page 56
BindDrivers Page 30
Block Size Page 56
Blocks Per Track, Definition TRM-44
Blocks Per Track Page 55
TRM-4
TRM-18
Blocks Page 55
TRM-4
TRM-18
Boot Disk Startup-Sequence Page 12
Page 33
Buffers TRM-18
BufferSize TRM-25
BuffMemType TRM-18
Bus Pass-Thru, A-1000 Page 9
TRM-37
Bus Termination TRM-34
Cable Options TRM-35
Calling Customer Service TRM-54
CAP TRM-5
TRM-12
TRM-16
TRM-56
Capacity, Calculating TRM-43
Capacity, SCSI device TRM-5
Capacity TRM-56
Check TRM-11
TRM-16
TRM-55
CheckFree TRM-11
TRM-13
Clear TRM-11
TRM-16
TRM-56
ClearSCSI TRM-11
TRM-16
TRM-56
CLtd-Format_0 TRM-12
TRM-16
CLtd-Format_1 TRM-12
TRM-16
CLtd-Format_2 TRM-12
TRM-16
CLtd-Format_3 TRM-12
TRM-16
CLtd.device TRM-9
TRM-11
Compuserve, C Ltd. Page 1
Copying Programs to Hard Drives Page 29
Copyright Notices Page 2
Customer Service TRM-54
Cylinders, Definition TRM-44

Index

Cylinders Page 55, TRM-4
C Ltd. Startup-Screen Page 12, Page 33
Data Transfer Rate TRM-49
Dealing with Static Electricity Page 4
DefDisk Page 33, TRM-11, TRM-13
Defects, Detection TRM-56
Defects, Mapping-out Page 57
Device TRM-18
Devices, Mountlist Page 39
DevInstall Page 31, TRM-11, TRM-13
DevSetup File Editor (MLed) Page 35
DevSetup File, Using the Page31
DevSetup TRM-11, TRM-13, TRM-19
DH0: - Logical Device TRM-40
DH1: - Logical Device TRM-40
DH2: - Logical Device TRM-40
DH3: - Logical Device TRM-40
DiskChange TRM-48
DMA Data Transfer TRM-49
DosType TRM-18
Editing the Bad Block List Page 57, Page 59
Editing the DevSetup File Page 35
Editing the MountList Page 35
Error Mapping Page 47, Page 56
Errors, Hard Drive, Bad Block List Page 57, Page 59
Errors, Hard Drive Page 42, Page 47, Page 56, Page 57
Errors TRM-12, TRM-17, TRM-56
FastFileSystem TRM-9, TRM-18
FAX, C Ltd. Page 1
FEdX (Low-Level Formatter) Page 44, Page 50, TRM-12, TRM-16
FFS TRM-9
File Server Network TRM-47
FileSystem TRM-18
Flags TRM-18, TRM-22, TRM-23, TRM-26, TRM-27, TRM-28
Floppy.Startup-Sequence TRM-11, TRM-15
Form Factor Page 14, Page 20
Format Page 62
Formatter TRM-12, TRM-17, TRM-57
Formatting, DOS Level Page 62
Formatting, General Page 41
Formatting, Low-Level Page 45
General Precautions Page 4
GenFormat TRM-12, TRM-17, TRM-57
GlobVec TRM-18
Glossary TRM-2
Guaranty, Hard Drive Mechanisms Page 3
Guaranty, SCSI Host/Controller Page 3
HandlerCode TRM-21
Handlers, DevSetup Page 39

Index

Hard Drive Back-Up Systems TRM-38
Hard Drive Bad Block List Page 57
Page 59
Hard Drive Bitmap TRM-47
Hard Drive Capacity, Reading TRM-5
Hard Drive Configuration Page 23
Hard Drive Errors Page 42
Page 47
Hard Drive Formatting Page 23
Hard Drive Icon Page 28
Hard Drive Installation
C Ltd.
General Page 11
External Page 13
Internal Page 14
non-C Ltd.
General Page 17
External Page 18
Internal Page 20
Configuration Page 23
Formatting Page 23
Hard Drive Organization TRM-44
Hard Drive Sharing TRM-47
Hard Drive Validation Page 33
HardDisk.driver TRM-9
TRM-11
TRM-13
HardDrive.Startup-Sequence TRM-11
TRM-14
Heads Page 55
Heads TRM-4
Help, FEdX, Defaults Page 52
Help, FEdX Page 53
Help, MLed Page 36
Page 40
Help TRM-52
HighCyl TRM-18
Host/Controller Installation
General Page7
A-500 Page 8
A-1000 Page 9
A-2000 Page 10
Host/Controller Model Numbers Page 5
HostAddress TRM-22
Icons TRM-53
Imbedded SCSI Hard Drives Page 14
Page 18
Page 20
Inquir TRM-12
TRM-16
TRM-55
Inquire TRM-12
TRM-12
TRM-16
TRM-55
Installation, Hard Drive
C Ltd.
General Page 11
External Page 13
Internal Page 14
non-C Ltd.
General Page 17
External Page 18
Internal Page 20
Configuration Page 23
Formatting Page 23
Installation, SCSI Host/Controller
General Page7
A-500 Page 8
A-1000 Page 9
A-2000 Page 10
Installing Programs onto Disk Page 29
Interleave Factor, Definition TRM-44
Interleave Factor, Suggested TRM-46
Interleave Page 42
Page 43
Page 56
TRM-18
IOmega Bernoulli Box Page 56
Logical Devices TRM-39
Logical Unit Number TRM-26
Low-Level Formatting Page 47

Index

LowCyl TRM-18
MacIntosh SCSI TRM-34
Matching Serial Numbers Page 6
Method of Data Transfer TRM-49
MLed Help Page 36
Page 40
MLed Loading Data Page 36
MLed Menues Page 36
MLed Saving Data Page 36
MLed Page 35
MLed TRM-11
TRM-15
Mount Page 30
TRM-18
Mounting a Hard Drive Page 30
MountList Editor (MLed) Page 35
MountList, Sample Partitioned TRM-40
MountList TRM-4
TRM-11
TRM-14
Multi-Host Systems TRM-7
Multi-User systems TRM-47
Multiple Devices, Eapansion Bus TRM-53
Multiple Partitions TRM-39
Network Termination TRM-34
Networking, SCSI TRM-47
Networking TRM-7
NewLook TRM-11
TRM-15
Non-SCSI Unit Number TRM-26
Number of Cylinders Page 55
TRM-4
Number of Heads Page 55
TRM-4
Opening The Hard Disk Page 28
Optical WORM Drives TRM-38
Other SCSI Devices (non-storage) TRM-51
P.D. Software on Hard Drives Page 28
Partition Size TRM-43
Partitioned Sample Mountlist TRM-40
Partitioning, Hard Drive TRM-39
Passing the Bus, A-1000 Page 9
TRM-37
Physical Device TRM-39
Pseudo-SCSI, definition TRM-2
Pseudo-SCSI TRM-34
Quick-Start,
General Page5
Controller Installation
General Page7
A-500 Page 8
A-1000 Page 9
A-2000 Page 10
Hard Drive Installation
C Ltd.
General Page 11
External Page 13
Internal Page 14
non-C Ltd.
General Page 17
External Page 18
Internal Page 20
Configuration Page 23
Formatting Page 23
ReadErr Page 42
Page 60
TRM-12
TRM-17
TRM-57
Removable Media SCSI Drives TRM-38
Reserved TRM-18
Ribbon Cable TRM-35
Sample DevSetup File TRM-19
Sample MountList TRM-18
SCSI Address TRM-7
TRM-22
SCSI Arbitration TRM-7
TRM-8
SCSI Bus, definition TRM-2
SCSI Bus TRM-34

Index

SCSI Controller Model Numbers	Page 5
SCSI Controller Power Required	TRM-49
SCSI Device Controller	TRM-2
SCSI Device, definition	TRM-2
SCSI format	TRM-57
SCSI Host Adapter, definition	TRM-2
SCSI Host Address	TRM-22
SCSI Host, definition	TRM-2
SCSI Host/Controller	TRM-3
SCSI Initiator, definition	TRM-3
SCSI Laser Printers	TRM-51
SCSI Network, definition	TRM-2
SCSI Network	TRM-34
SCSI Operation, Overview	TRM-6
SCSI Page Scanners	TRM-51
SCSI programming	TRM-5
SCSI Tape Drives	TRM-38
SCSI Target, definition	TRM-3
SCSI Termination	TRM-34
SCSI to ST-506 Adapter	Page 14
	Page 18
	Page 20
SCSI Unit Address	TRM-25
	TRM-33
SCSI Unit Number	TRM-7
	TRM-25
SCSI, definition	TRM-2
SCSI, Overview	TRM-4
SCSI-Link Library	TRM-5
	TRM-11
	TRM-14
	TRM-50
	TRM-51
	TRM-58
SCSI.handler	TRM-9
	TRM-11
	TRM-13
SCSICheck	TRM-12
	TRM-16
	TRM-55
SCSIdos 3.0, Overview	TRM-9
SCSInet	TRM-7
	TRM-47
Sector Errors	TRM-56
Serial Numbers, Matching of	Page 6
Setting Interleave	TRM-44
Setting the SCSI Address	TRM-33
Shielded SCSI Cable	TRM-35
Size	TRM-12
	TRM-16
	TRM-56
Software on C Ltd. Hard Drives	Page 28
Specificationss, Host/Controller	TRM-49
ST-506 Hard Drives	Page 14
	Page 18
	Page 20
StackSize	TRM-18
	TRM-21
	TRM-25
Startup-Sequence	TRM-11
	TRM-14
Static Electricity	Page 4
Suggested Interleaves	TRM-46
Surfaces, Definition	TRM-44
Surfaces	TRM-18
Tape Back-up	TRM-38
TaskName	TRM-21
	TRM-25
TaskPriority	TRM-21
	TRM-25
Termination Resistors	TRM-34
Termination, definition	TRM-2
Tracks, Definition	TRM-44
Trademark Notices	Page 2
Trouble	TRM-52
typenote	TRM-11
	TRM-15
Unit LUN	TRM-26
Unit Number	Page 56
Unit	TRM-18